LEVEL

SAMSO-TR-79-25

BRASSBOARD FAULT TOLERANT SPACEBORNE COMPUTER (BFTSC)

12

NW

FINAL REPORT

VOLUME 1 OF 3

20 DECEMBER 1978

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

Prepared For

SPACE AND MISSILE SYSTEMS ORGANIZATION
AIR FORCE SYSTEMS COMMAND
LOS ANGELES AIR FORCE STATION
BOX 92960, WPC
LOS ANGELES, CALIFORNIA 90009

Prepared By

RAYTHEON

RAYTHEON COMPANY
EQUIPMENT DIVISION
STRATEGIC SYSTEMS DIRECTORATE
SUDBURY, MASSACHUSETTS

41172

Volume I (of three) of this final report was submitted by Raytheon Co., Equipment Division, Strategic Systems, Sudbury, Massachusetts, under contract F04701-75-C-0149 with the Space and Missile Systems Organization, Los Angeles Air Force Station, Los Angeles, California. Captain Roy L. Schmiesing (YCD) is the Project Manager.

This report has been reviewed by the Information Office (OI) and is releasable to the National Technical Information Services (NTIS). AT NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication. Publication of this report does not constitute Air Force approval of the report's findings or conclusions. It is published only for the exchange and stimulation of ideas.

Roy L. Schmiesing

ROY L. SCHMIESING, CAPT, USAF
Project Manager
Advanced Space Computer Division

Ronald G. Spray

RONALD G. SPRAY, LT COL, USAF
Deputy Director for Subsystems

FOR THE COMMANDER

Leonard E. Baltzell

LEONARD E. BALTZELL, Col, USAF
Asst. Deputy for Advanced
Space Programs

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER: BFTSC - A021	
4. TITLE (and Subtitle): Brassboard Fault Tolerant Spaceborne Computer (BFTSC), Volume I (Of three)	5. TYPE OF REPORT & PERIOD COVERED: Final Report, Volume I, 15 May 75 — 14 May 79
	6. PERFORMING ORG. REPORT NUMBER: ER78-4457-Vol-1
7. AUTHOR(s): A.H. VanDoren	8. CONTRACT OR GRANT NUMBER(s): F04701-75-C-0149 new
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Raytheon Co., Equipment Division, Strategic Systems Dir., Sudbury, Massachusetss 01776	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: PE63401F, Project 2181
11. CONTROLLING OFFICE NAME AND ADDRESS: Space and Missile Systems Orgn. (YCD), P.O. Box 92960, Worldway Postal Center, Los Angeles, California 90009	12. REPORT DATE: 20 December 1978
	13. NUMBER OF PAGES: 177
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): 172p.	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

SAMSO Project Manager: Captain Roy L. Schmiesing (YCD)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Fault Tolerant Computing — Reliable Computing

Spacecraft Computer — Radiation Hardened Computer

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This Report describes the design and development of the Brassboard Fault-Tolerant Spaceborne Computer (BFTSC) for the United States Air Force, Space and Missile Systems Organization. The BFTSC is a breadboard model of a self-repairing, highly reliable, general-purpose computer designed to have a 95 percent probability of surviving a five-year mission in a space environment without any degradation in performance. The Report is written in three volumes. Vol I contains a description of the computer architecture and the methods employed in achieving Fault Tolerance, with a focus on redundancy techniques and fault detection(over)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

411712

mechanisms. Vol. II describes the architectural trade-offs and the fault recovery hardware/software interaction. It also documents the BFTSC construction and testing results, as well as the recommended technologies for fabricating a flight unit computer. Vol. III describes work directed specifically toward a prototype flight unit.

RAYTHEON COMPANY
EQUIPMENT DIVISION

BRASSBOARD FAULT TOLERANT SPACEBORNE COMPUTER (BFTSC)

(F04701-75-C-0149)

FINAL REPORT

VOLUME I

(CDRL A021)

ER78-4457

20 DECEMBER 1978

Prepared For

DEPARTMENT OF THE AIR FORCE
HEADQUARTERS SPACE AND MISSILE SYSTEMS ORGANIZATION (AFSC)
P.O. BOX 92960, WORLDWAY POSTAL CENTER
Los Angeles, California 90009

Prepared By

RAYTHEON COMPANY
EQUIPMENT DIVISION
Strategic Systems Directorate
528 Boston Post Road
Sudbury, Massachusetts 01776

PREFACE

The final report for the Brassboard Fault Tolerant Spaceborne Computer is being prepared in three volumes. Volume 1 contains a description of the computer architecture and the methods employed in achieving fault tolerance. Volume 2, when issued, will contain a description of the fault recovery process, and an overview and listing of key BFTSC documents. Primarily as it is planned for the prototype FTSC, Volume 3, when issued, will contain the results of the post-brassboard activity, through the end of contract.

CONTENTS
VOLUME I

CONTENTS (Continued)

CONTENTS (Continued)

ILLUSTRATIONS
VOLUME I

TABLES
VOLUME I

SECTION 1

INTRODUCTION

The Brassboard Fault-Tolerant Spaceborne Computer (BFTSC) is the breadboard of a self-healing, highly reliable, general-purpose computer designed to have a 95 percent probability of surviving a five-year mission in a space environment without any degradation in performance. The BFTSC is the result of a 17-month hardware design and development effort by Raytheon Company for the United States Air Force, Space and Missile Systems Organization. The computer is the predecessor of the flyable prototype Fault-Tolerant Spaceborne Computer (FTSC).

The BFTSC is a high-reliability general purpose micro-programmed computer with the following features:

COMPUTATIONAL

- 100,000 operations/second (>200,000 in the FTSC)
- 95 instructions (expandable to 128)
- Integer, floating point, and vector operations
- Eight general purpose registers
- Four working registers
- Four special purpose registers
- 32-bit data
- 61,440 words of addressable memory

RELIABILITY

- Demonstrates fault-tolerant concepts necessary to achieve >95 percent probability of operating for five years

I/O

- One high-speed redundant DMA channel (two in the FTSC)
- Three low-speed redundant user channels (up to 62 in the FTSC)

The computer is segregated into functional elements with each element protected by at least one spare. In addition, several of these elements, notably the memory elements and the address and data buses, are provided with

redundancy at the subelement level through the use of rippler switches, thereby significantly reducing the redundant hardware that would otherwise be required. The Configuration Control Unit (the ultimate guardian of the system configuration) has been reduced in complexity by distributing much of its function (and hence its unreliability) to other computer elements. High fault coverage is assured through the extensive use of coding, and single point failure mechanisms have been avoided by, among other things, providing at least two paths for all critical signals.

For the purpose of clarity of this document, a Glossary of Terms is provided as Appendix B of this document.

SECTION 2

APPLICABLE DOCUMENTS

CDRL	Title
A006	Subsystem Design Analysis Report, BFTSC Architecture
A007	Configuration Item Development Specification, BFTSC Architecture Design
A010	Configuration Item Product Fabrication Specification, BFTSC Logic Design
A011	Computer Program Development Specification, BFTSC Executive/Recovery Software Design
A014	Technical Data for FTSC Prototype Specification
A016	Executive/Recovery Software User's Manual
A017	Computer Program Product Specification Executive/Recovery Software
A029	Prototype Enhancement Recommendations
A030	Operations and Program Manual

SECTION 3

ARCHITECTURAL OVERVIEW

The Brassboard Fault-Tolerant Spaceborne Computer (BFTSC) is a general purpose, stored program, 32-bit, parallel computer capable of executing a repertoire of instructions in a variety of addressing modes. The BFTSC processing characteristics are specified in Table 3-1. The BFTSC modules and their interconnection buses are as shown in Figure 3-1. Each module within the BFTSC contains an interface connecting it to the internal buses. All intermodule communication in both normal and reconfiguration modes takes place via these buses.

3.1 BUS DESCRIPTION

There are seven internal buses in the BFTSC (shown in Figure 3-1), i.e., Address Bus, Data Bus, Control Bus, Interrupt Bus, Status Bus, Timing Bus, and Power Bus. Each of these buses is either provided with switchable spare lines or implemented in a triple-modular-redundant (TMR) configuration. The function of each of these buses is described below:

a. Address Bus - Contains the lines which convey the 16 bits comprising each address plus the eight bits comprising its error code.

b. Data Bus - Contains the lines which convey the 32 bits comprising each data word plus the eight bits comprising its error code.

c. Control Bus - Contains the lines which control bus access and the lines indicating the beginning, end, direction, and type of transfer.

d. Interrupt Bus - Contains the lines indicating interrupts from each of the DMAs, SIUs, Timing Modules, and CCUs.

e. Status Bus - Contains the lines indicating out-of-tolerance conditions in the Timing and Power Units, the lines which indicate detected faults, and the lines which convey configuration status information.

f. Timing Bus - Connects the system clock lines and the Hardened Timer outputs to the rest of the system.

g. Power Bus - Connects the system voltages from the Power Modules to the rest of the system.

Table 3-1. BFTSC Characteristics

Type:	General purpose, stored program, parallel, digital computer
Arithmetic	Binary, two's complement, fixed-point and floating-point
Data Word Formats:	32-bit fixed-point (integer) 32-bit floating-point (24-bit mantissa and 8-bit radix-2 exponent)
Instruction Word Format:	Single 32-bit format
Address Modes:	Register-Register Immediate Direct Indirect Indexed, Postincrement Indexed, Predecrement Indexed Indexed, Indirect
Architecture:	Integrated floating-point and fixed-point (integer) processor; microprogrammed centralized control
Registers:	Eight general purpose registers (32-bit) that serve as accumulators, index registers, or address pointers Eight working and special purpose registers that include dedicated program registers as well as data and memory address registers
Interrupts:	Eight normal interrupts, an illegal op code interrupt, and a fault interrupt

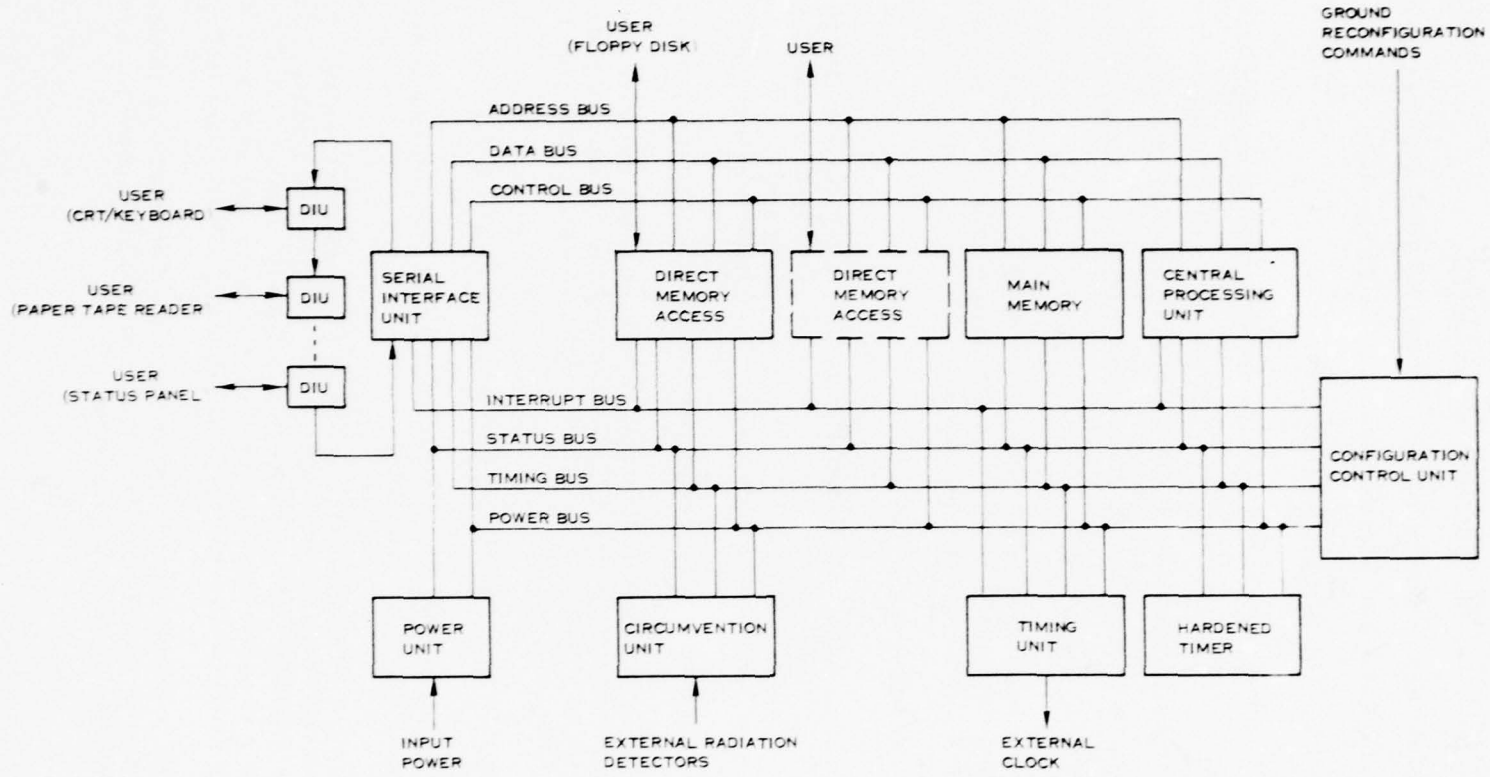


Figure 3-1. BFTSC System Block Diagram

3.2 DESCRIPTION OF MODULE FUNCTIONS

Functional descriptions of each of the BFTSC modules are contained in the following paragraphs and are summarized in Table 3-2.

3.2.1 CENTRAL PROCESSING UNIT (CPU)

The CPU implements all arithmetic and logical operations required to process application and fault recovery programs. There are four CPU Modules in the BFTSC, two of which are powered during normal operation. One of the powered CPUs is designated as the active module, the other as the monitor. Each CPU has the following features:

- Executes the 95 instructions as defined in Appendix A. These instructions are performed on 32-bit data words and include integer, floating point, and logical operations.
- Eight addressing modes:
  - Register-Register
  - Immediate
  - Direct
  - Indirect
  - Indexed, Postincrement
  - Indexed, Predecrement
  - Indexed
  - Indexed, Indirect
- Microprogrammed using a 1K x 80-bit read-only memory (ROM).
- Eight program addressable 32-bit general purpose registers, an extension register, a program counter, status registers, a memory address register, a memory data register, and four working registers.
- Capable of recognizing and processing ten levels of interrupts.
- Incorporates a bus arbiter to resolve bus access conflicts among the CPU, SIU, and DMA.
- Capable of operating in either of two modes: active or monitor. The active CPU provides all information and control to the system; the monitor CPU compares the outputs from the active CPU with those it would have generated if it were the active CPU and reports any discrepancy to the Configuration Control Unit (CCU).
- A 1K x 32-bit reconfiguration read-only memory (ROM) which contains the reconfiguration program. This program is initiated in response to a fault interrupt.

Table 3-2. BFTSC Functional Units

Function	Description
CPU - Central Processing Unit	The CPU implements all arithmetic and logical operations required to process BFTSC programs.
MM - Main Memory	The MM contains all active BFTSC programs and data.
CCU - Configuration Control Unit	The CCU coordinates hardware configuration management in the BFTSC. Upon detection of a fault within the system, it selects a CPU and bus configuration and initiates the recovery program. The CCU supplies the necessary hardware status information to the recovery program for successful diagnosis of the failure.
DMA - Direct Memory Access	The DMA allows a peripheral device to transfer data directly to MM at rates of up to 5M bits/second. Data transfers are initiated under program control, but are performed independently of the CPU.
SIU - Serial Interface Unit	The SIU is the connection with the serial data bus that allows up to 62 low-speed peripheral units to interface to the BFTSC in a direct memory access mode.
DIU - Device Interface Unit	The DIUs interface peripheral devices to the serial data bus at rates of up to 325K bits/second. The DIUs are installed within the peripheral housing.
TU - Timing Unit	The TU provides the BFTSC clock and the real-time interrupt.
PU - Power Unit	The PU supplies all regulated power required to operate the BFTSC.
HT - Hardened Timer	The HT accumulates elapsed time during recovery periods and radiation events so that programs which are time dependent can be restarted without degradation after recovery has been effected.
CU - Circumvention Unit	The CU monitors radiation levels to determine when the BFTSC will cease to operate correctly. When this radiation level occurs, memory and I/O clamps are activated to protect the memory contents and prevent I/O changes.

3.2.2 CONFIGURATION CONTROL UNIT (CCU)

The CCU has the capability to interrupt the CPU whenever a fault is detected and to provide sufficient status information to ensure a successful reconfiguration following any non-catastrophic fault. There are three CCU Modules in the BFTSC which operate in a triple-modular-redundant (TMR) configuration. Each CCU performs the following functions:

- Causes the fault recovery program to be initiated by issuing a fault interrupt to the CPU whenever a failure is detected by one or more of the fault detection monitors. (The fault interrupt is not issued if the specific failure detected has been previously masked under program control.)
- Changes the CPU, address bus, and/or data bus configurations whenever a failure indicates that such a change is necessary to effect a successful recovery.
- Accepts program flag inputs from the CPU. The functions performed by these inputs include masking certain types of faults and causing the CPU, address bus, or data bus states to be changed.

3.2.3 MEMORY MODULE (MM)

The main memory in the BFTSC is composed of three independent Memory Modules each with a capacity of 4096 words. The design permits expansion (in the FTSC) to 24 Memory Modules, 15 of which can be active, directly addressable memory (61,440 words). The remaining Memory Modules are unpowered spares. Each Memory Module has the following features:

- Provides non-volatile, non-destructive readout, random access storage for BFTSC programs and data.
- A word size of 41 bits, consisting of 38 active bits and three spare bits. The 38 active bits consist of 32 bits of data and six bits of code.
- The capability to select the active and spare bit-line positions under program control. This selection applies to all words in the module so that if a bit is identified as a spare, that bit is a spare in each word of the module.
- May be assigned any one of the following 15 sets of contiguous addresses: 0 to 4K-1, 4K to 8K-1, ... , 56K to 60K-1 (K indicates 1024 words).

- Ability to individually write-protect each memory subblock under program control. (Each Memory Module is divided into four subblocks of 1024 contiguous locations each.) Any attempt to store into a write-protected area leaves the information in that address unchanged and causes a fault.

- Capability of detecting multiple bit errors using the 6-bit code in each word. The memory can also correct single bit errors when put in the error correcting mode under program control.

- Can be powered on or off under program control. (Following power-on, a Memory Module does not respond to read or write operations until so directed under program control.)

3.2.4 SERIAL DATA BUS

The serial data bus subsystem allows up to 62 external interfaces to the system. Three are provided in the BFTSC; they connect to the keyboard/CRT, the paper tape reader, and the status panel. The serial bus is controlled by the Serial Interface Unit (SIU) and is connected to the peripherals through dual-redundant Device Interface Units (DIU). It is a uni-directional, daisy-chain system with dual-redundant serial links between each contiguous pair of DIUs along the bus.

3.2.4.1 Serial Interface Unit (SIU)

The BFTSC contains two SIU Modules connected in a dual-redundant configuration. Only one SIU is powered at a time; it can be powered off and the other powered on under program control when a failure is detected. Each SIU Module has the following capabilities:

- Transfers blocks of words between the serial bus and main memory using a direct-memory-access capability.

- Is able to sequentially access any number of locations in a program specified block (module) of memory. It fetches a word from each address, transmits the word over the serial bus, and stores the returned (possibly modified) word back into the memory location from which it was fetched.

- Synchronizes the initiation of a block transfer with the real-time interrupt (RTI). (In the Immediate Mode (cf. Section 5.3.4.1), the block transfer is initiated with the next transmitted word.) A new block transfer can not be initiated if the SIU is already in the process of transmitting a block.

- Can generate two levels of interrupts: an end-of-block interrupt signalling the completion of the current block transfer, and a high-priority interrupt indicating peripheral initiated interrupts or a failure condition.

3.2.4.2 Device Interface Unit (DIU)

Each peripheral in the BFTSC is connected to the serial bus through a dual-redundant pair of DIU Modules. Both DIUs in each pair are powered at all times by the peripheral with which they are associated. Each DIU has the following characteristics:

- Transmits or receives status (command) information or data in accordance with the op code field of only those received words containing its unique pin programmed address.
- Following a peripheral initiated interrupt, the module modifies the first available serial bus word to indicate an interrupt to the SIU (unless a higher priority interrupt is already indicated in the word). The DIU repeats the interrupt every third real-time interrupt (RTI) until it is acknolwedged.

3.2.5 DIRECT MEMORY ACCESS (DMA)

The DMA provides a high speed parallel interface to the BFTSC. There is one dual-redundant pair of DMA modules in the BFTSC (two pairs in the FTSC) which is connected to the floppy disc. Only one DMA of the pair is powered on at a time: the other acts as an unpowered spare and can be powered on or off under program control when a failure is detected. Each DMA module is capable of the following:

- Transferring blocks of information directly from any active Memory Module to the peripheral or from the peripheral to memory.
- Transferring blocks initiated under program control without CPU intervention.
- Passing on commands sent by the CPU to control the peripheral.
- Generating two levels of interrupts: an end-of-block interrupt signalling the completion of the current block transfer, and a high-priority interrupt indicating a peripheral-initiated interrupt or a failure condition.

3.2.6 TIMING UNIT (TU)

The Timing Unit consists of two powered Timing Modules in a dual-redundant configuration. One module is active, while the other serves as a standby spare. Each Timing Module has the following features:

- When active, it generates a system clock at a frequency of 1.25 MHz (2.5 MHz in FTSC).
- When active, it generates an internal real-time interrupt (RTI) every 8.8064 ms. Capability is provided to decrease this period by a factor of 2 or to increase it by a factor of 2 or 4. (All the times are divided by 2 in the FTSC.)
- When active, it generates an external real-time clock at a frequency of 312.5 kHz.
- Provides dual status signals to other BFTSC modules to indicate its active/standby state.
- Dual monitors to check its system clock and internal real-time interrupt outputs. If an out-of-tolerance condition is detected, the detecting Timing Module enters the standby state and the other Timing Module takes over the active role.

3.2.7 POWER UNIT (PU)

The Power Unit consists of two Power Modules connected in a dual-redundant configuration. One module is active while the other serves as a powered standby spare. In the BFTSC, one commercial supply for each internal voltage supplies both Power Modules. (In the FTSC, each Power Module shall convert the primary power input to the internal levels required.) Each Power Module has the following capabilities:

- When active it provides the internal voltages required by the BFTSC.
- Provides dual status signals to the other BFTSC modules to indicate its active/standby state.
- Contains dual monitors to examine its output voltage levels. If an out-of-tolerance condition is detected, the detecting Power Module enters the standby state and the other Power Module takes over the active role.

3.2.8 CIRCUMVENTION UNIT (CU)

The Circumvention Unit is designed to protect the BFTSC during radiation events, power interruptions, and timing interruptions by clamping all Memory, DMA, and SIU Modules. There are three Circumvention Modules in the BFTSC operating in a triple-modular-redundant (TMR) configuration. Each Circumvention Module has the following characteristics:

- Accepts inputs from distributed radiation detectors. These inputs can be connected to a simulated radiation source in the BFTSC.
- Monitors the status signals indicating out-of-tolerance power or timing conditions.
- During the period following detection of radiation, power, or timing out-of-tolerance conditions, generates grounded clamp signals which prevent memory contents from being changed and inhibit all I/O.

3.2.9 HARDENED TIMER (HT)

The Hardened Timer keeps track of real time during radiation events, power module failures, timing module failures, and reconfiguration after a fault. There are three Hardened Timer Modules in the BFTSC operating in a triple-modular-redundant (TMR) configuration. Each Hardened Timer Module is capable of the following:

- Being activated upon the occurrence of a radiation event, a power failure, a timing failure, or the detection of any unmasked fault by the CCU. (It receives power from the energy storage unit so that it can operate during a power failure.)
- Accumulating the number of real-time interrupts (RTI) which occur during the time in which the timer is active by charging a capacitor from an internal linear current source.
- Stopping accumulation of time under program control.
- Reading and resetting the accumulated time contents under program control.
- Accumulating up to 6 seconds of elapsed time.

SECTION 4

REDUNDANCY TECHNIQUES

Three redundancy techniques are employed within the BFTSC: triple-modular redundancy, spare module redundancy, and subelement redundancy. The BFTSC is partitioned such that each functional element utilizes one or more of these techniques to obtain the required reliability for that element with a minimum of redundant hardware.

4.1 TRIPLE-MODULAR REDUNDANCY (TMR)

An element which is implemented in a triple-modular-redundant (TMR) configuration consists of three identical, independent elements simultaneously performing the same function. The outputs from these three elements are subjected to a majority vote wherever they are used. As a result, the failure of a single element is effectively masked. In the BFTSC, triple-modular redundancy is used for the Configuration Control Unit (CCU), the Circumvention Unit (CU), the Hardened Timer (HT), and some of the Control Bus signals (cf. Section 5.1.3).

4.2 SPARE MODULE REDUNDANCY

Several elements within the BFTSC have redundancy at the module level; an entire module can be directly replaced by a spare. This type of redundancy is used in the Direct Memory Access Units (DMA), Serial Interface Unit (SIU), Device Interface Units (DIU), Power Unit (PU), Timing Unit (TU), Central Processing Unit (CPU), and Memory Module (MM).

Each DMA and SIU consists of a dual-redundant pair of DMA or SIU Modules. Only one module of each pair is active with the other serving as an unpowered spare. When a failure is detected in the active module, it is powered off and its spare is powered on under program control. Each DMA Module within the pair has its own external interface so that redundancy can be extended through to the user. Each SIU Module transmits on a different serial bus line.

The DIUs each consist of a dual-redundant pair of DIU Modules, both of which are powered at all times. Each DIU Module can receive from either of the serial bus lines from the previous pair on the bus, but it transmits on only one line.

An operational serial bus configuration is ensured, by the self-switching serial bus monitors within the following DIU Modules (cf. Sections 6.9 and 6.10), for any failure within a DIU Module or the serial bus line on which it transmits. Each DIU Module, within a pair, has its own external interface allowing the redundancy to be extended to the user.

The Power Unit and Timing Unit both consist of dual-redundant pairs of Power or Timing Modules. One of the modules in each pair is active and the other serves as a standby spare. When a failure occurs in the active module, the dual-monitors in each module (cf. Section 6.22 and 6.23) automatically switch the spare to the active state.

There are four CPU Modules in the BFTSC, any two of which can be powered at a time. One of the two powered CPUs is active while the other serves as a monitor (cf. Section 5.3.1). When a failure occurs in one of the powered CPU Modules, a different combination of CPU Modules is powered on. Pairs of CPU Modules are tested in a predetermined cyclic sequence until an operational pair is found. As long as at least two CPU Modules are functional, an operational CPU configuration will be found.

The Main Memory is divided into Memory Modules each containing 4096 (4K) words of data storage. Up to 24 Memory Modules can be accommodated by the BFTSC, 15 of which can be active. The remaining Memory Modules serve as unpowered spares. Each active Memory Module is assigned a "soft name", under program control, which determines the 4K block of addresses to which that Memory Module responds. When one of the active Memory Modules fails, a spare is powered on and assigned the same soft name under program control while the failed module is turned off.

4.3 SUBELEMENT REDUNDANCY

Several elements within the BFTSC are designed to take advantage of the generally unappreciated fact that the amount of improvement to be gained in the reliability of any element from a given expenditure in hardware is, at least up to a point, an increasing function of the number of identical subelements into which it can be segregated.

Figure 4-1 illustrates this concept. Figure 4-1(a) symbolically represents an element having a mean time before failure (MTBF) of one year. The probability P_s that such an element operates without failure for a period of one year is, under the usual constant failure rate assumption, only 0.37. If the same element is provided with a standby spare (Figure 4-1(b)) which can substitue for it should the original element fail, and assuming the switch itself does not fail, one can easily verify that this 100 percent hardware increase results in a 50 percent increase in MTBF and a 62 percent increase in P_s. On the other hand, if the original element could be segmented into four identical subelements, and a single spare were provided which could be substituted for any one of these subelements (Figure 4-1(c)), the resulting 25 percent increase in hardware would increase the MTBF by 80 percent over that of the original non-redundant element and would increase P_s by 86 percent. Thus, a 25 percent hardware increase at the subelement level is theoretically more effective than a 100 percent increase at the element level. Or, if a 100 percent hardware increase were applied at the subelement level, as shown in Figure 4-1(d) with any one of the spare subelements able to assume the role of any one of the original subelements, the increase in the MTBF and in P_s would be 254 percent and 166 percent, respectively, a major improvement over that afforded by applying the same percentage hardware increase at the element level.

The conclusions to be drawn from this example, however, are subject to one very important qualification: the switch inevitably needed to accomplish the desired substitutions will itself be subject to failure. Since the reliability of the switch must, in fact, decrease as the number of subelements to be switched increases, the point must be reached (were it possible to subdivide the element indefinitely) at which the switch itself becomes the dominant source of failure. Obviously, it would not be profitable to subdivide the element further once this point is reached, even if it were physically possible to do so.

One of the major obstacles in realizing the potential of the subelement redundancy technique is the need for a switch that can configure the various subelements into an operational system without simultaneously dissipating much of the promised gain in reliability. A switching device, called a "rippler switch" was designed to overcome this obstacle. Basically, the rippler functions by replacing any defective subelement in a linear array of identical devices by its

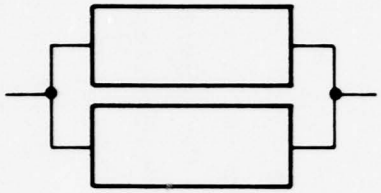
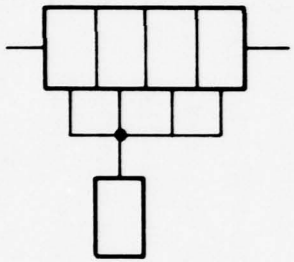
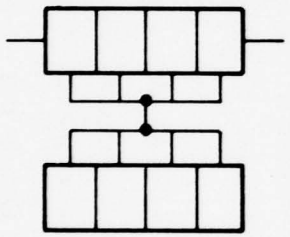
RELIABILITY BLOCK DIAGRAM	MTBF	P_S	HARDWARE INCREASE	COMMENT
(a) 	1 YEAR	37 %	0 %	NON–REDUNDANT
(b) 	1.5 YEARS	60 %	100 %	DUAL REDUNDANT
(c) 	1.8 YEARS	69 %	25 %	1 SPARE ON 4 SUBELEMENTS
(d) 	3.54 YEARS	98.6 %	100 %	4 SPARES ON 4 SUBELEMENTS

Figure 4-1. Redundancy Configurations

nearest neighbor (in some specified ordering), then replacing that device by its nearest neighbor, etc., the whole process "rippling" down the array of subelements until the last active device is replaced by the first available spare. The advantage of this switching method over the more conventional direct substitution approach is in its amenability to relatively simple (and hence reliably implemented) control algorithms.

Figure 4-2 demonstrates how the rippler accomplishes the desired switching for an element which is partitioned into five identical subelements with three spare subelements. Each subelement has associated with it a rippler slice which can assume any one of five states. The rippler states establish the data paths (indicated by the lines in Figure 4-2) which provide the link between the operating subelements and a set of input/output ports (represented by the numbers 1 through 5) thereby defining the functional role of each subelement. If the rippler slice associated with subelement i is in state R_0, the subelement i is connected to function i, if in state R_1, then subelement i is connected to function i-1. Similarly, in states R_2 and R_3, subelement i is connected to functions i-2 and i-3, respectively. Finally, in state R_*, subelement i is not connected to any function. Initially, each rippler slice is in state R_0 and the data paths are as shown in Figure 4-2(a).

Now, suppose subelement S_4 fails. The fourth rippler slice is set to R_* and all subsequent rippler slices are advanced one state to R_1. As shown in Figure 4-2(b), the defective subelement is rippled out and the first spare subelement is used. Similarly, Figure 4-2(c) shows the response to a second failure, this one in subelement S_2. The second rippler slice is set to state R_*, the third rippler slice advances one state to R_1, the fourth slice remains in state R_*, and the subsequent slices are advanced one state to R_2. As a result both failed subelements are rippled out and the first two spare subelements are rippled in. It is readily seen that this control algorithm will always result in an operational system, regardless of the order in which the failures occur, so long as the number of failures does not exceed the number of available spares.

In the BFTSC, subelement redundancy is employed for the address and data buses and for the memory bit lines in each memory module. The address bus (cf. Section 5.1.1) is partitioned into three 8-bit bytes with one spare byte, and the data bus (cf. Section 5.1.2) is partitioned into five 8-bit bytes with one spare byte. Each memory module (cf. Section 5.3.3) is partitioned into 38 bit lines with 3 spare bit lines.

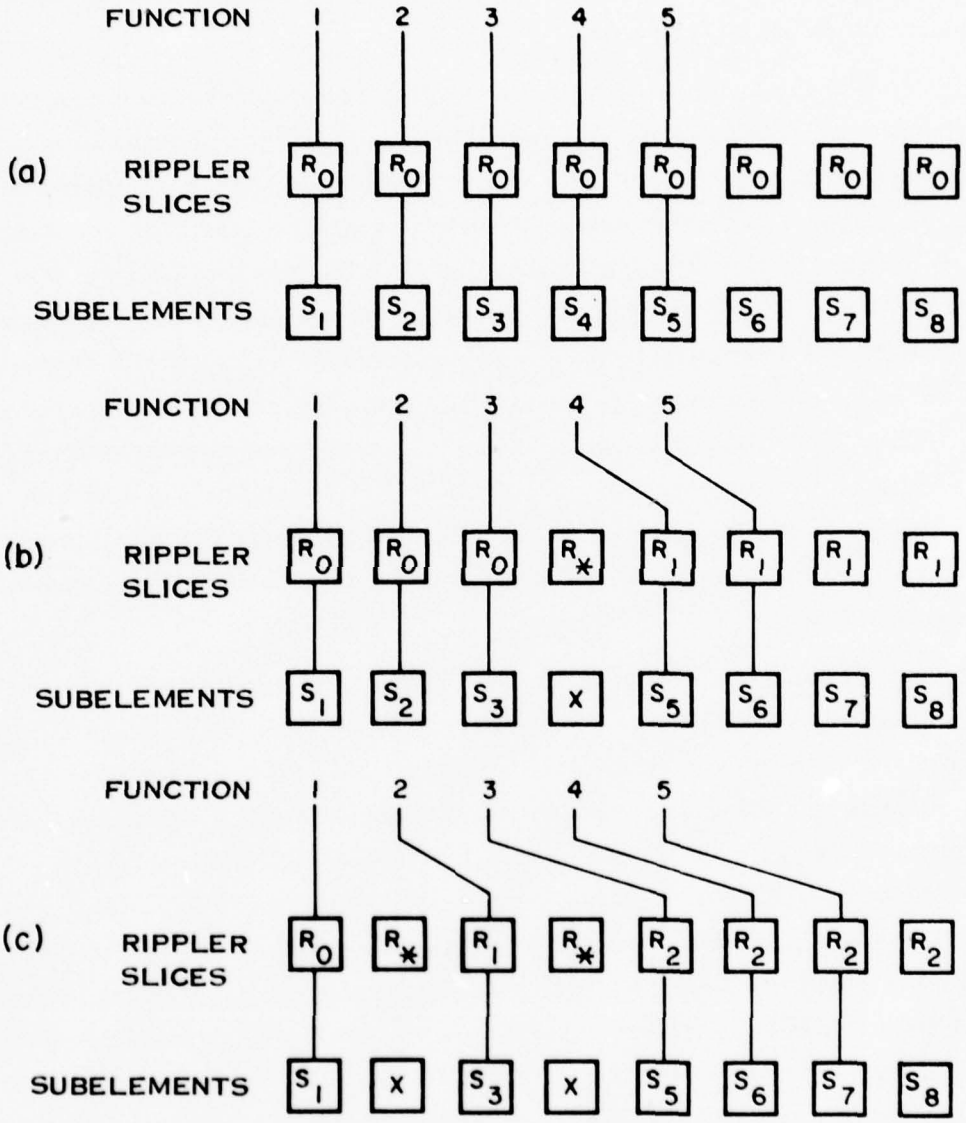


Figure 4-2. Typical Rippler Sequence

SECTION 5
DETAILED ARCHITECTURE

The Brassboard Fault-Tolerant Spaceborne Computer (BFTSC) is a high reliability, general purpose, microprogrammed, 32-bit computer with the processing characteristics listed in Table 3-1. The system modules and the buses which interconnect them are shown in Figure 3-1. Each module within the BFTSC contains an interface connecting it to the internal buses. All inter-module communication in both normal and reconfiguration modes takes place via these buses.

5.1 INTERNAL BUSES

The seven internal buses within the BFTSC have been configured to provide a sufficient level of redundancy to meet the reliability requirements of the computer. Each of the buses and its configuration is listed in Table 5-1.

Table 5-1. Internal Buses

Bus	Signals	Configuration
Address	Address Bits	Four 8-bit bytes: two bytes of address, one byte of code, and one spare byte. Bused among all CPU, Memory, DMA, and SIU Modules.
Data	Data Bits	Six 8-bit bytes: four bytes of data, one byte of code, and one spare byte. Bused among all CPU, Memory, DMA, and SIU Modules.
Control	Bus Access Request	One line from each DMA and SIU Module to all four CPU Modules. (cf. Figure 5.1)
	Bus Access Acknowledge	One line from all four CPU Modules to each DMA and SIU Module. (cf. Figure 5.1)
	Data Request	TMR: Three lines in a triple-modular-redundant configuration, generated in triplicate, voted at receiver. Bused among all CPU, Memory, DMA, and SIU Modules.

Table 5-1. Internal Buses (Continued)

Bus	Signals	Configuration
	Data Acknowledge	TMR: Generated in triplicate, voted at receiver. Bused among all CPU, Memory, DMA, and SIU Modules.
	Read/Write	TMR: Generated in triplicate, voted at receiver. Bused among all CPU, Memory, DMA, and SIU Modules.
	Hard/Soft	TMR: Generated in triplicate, voted at receiver. Bused among all CPU, Memory, DMA, and SIU Modules.
Interrupt	Fault Interrupt	TMR: One line from each CCU Module, voted at receiver. Bused among all CPU Modules.
	I/O Module Interrupts	Two interrupts from each DMA and SIU Module. Bused among all CPU Modules.
	Real-Time Interrupt	One line from each Timing Module; bused in a loop to all CPU and SIU Modules and then returned to the Timing Module.
Status	Power Status	Two lines from each Power Module; bused in a loop to all applicable modules and then returned to the Power Module.
	Timing Status	Two lines from each Timing Module, bused in a loop to all applicable modules and then returned to the Timing Module.
	Clamps	TMR: Each Circumvention Module generates three clamp signals (CCU clamp, I/O clamp, and Memory clamp); each clamp is voted at its receiver. The CCU clamp is bused among all CCU Modules; the I/O clamp is bused among all Memory, DMA, SIU and HT Modules; and the Memory clamp is bused among all Memory Modules.

Table 5-1. Internal Buses (Continued)

Bus	Signals	Configuration
	Reconfiguration Program Flags	Three lines generated by each CPU. Bused among all CCU Modules.
	Reconfiguration Status	TMR: Each of the three signals (alexic, reconfiguration mode, and reconfiguration mode with flag 1 not set) is generated in each CCU Module and voted at its receiver. Bused among all applicable modules.
	Fault Indication Type	Three lines (generated in triplicate) for each fault indication type bused among all sources of the fault. One line goes to each CCU Module.
	Fault Indication Location	Three lines (generated in triplicate) for each of the following module types: CPU, DMA1, DMA2, and SIU. Bused among all modules of that type. One line goes to each CCU Module.
	CPU Active/ Monitor Select	TMR: Four lines from each CCU Module, voted at receiver. Bused among all CPU Modules.
	Address Bus Rippler State	TMR: Two lines from each CCU Module, voted at receiver. Bused among all CPU, Memory, DMA, and SIU Modules.
	Data Bus Rippler State	TMR: Three lines from each CCU Module, voted at receiver. Bused among all CPU, Memory, DMA, and SIU Modules.
	Hardware Status	TMR: Eleven lines from each CCU Module, voted at receiver. Bused among all CPU Modules.
Timing	System Clock	One line from each Timing Module; bused in a loop to all applicable modules and then returned to the Timing Module.

Table 5-1. Internal Buses (Continued)

Bus	Signals	Configuration
	Hardened Timer Output	TMR: One line from each Hardened Timer Module, voted in software. Bused among all CPU Modules.
	Hardened Timer Reset	TMR: Generated in triplicate by active CPU, voted at receiver. Bused among all CPU and Hardened Timer Modules.
Power	System Voltages	One line for each voltage from each Power Module; bused in a loop to all applicable modules and then returned to the Power Module.
	Energy Storage	Two lines from each Power Module; bused among all Timing, Hardened Timer, and Circumvention Modules.

5.1.1 *ADDRESS BUS*

Each 16-bit address is encoded by appending to it the eight parity bits determined by a shortened cyclic code structure. The cyclic code used is that defined by the generator polynomial, $g(x) = (x^6 + x + 1)(x^2 + x + 1)$.

The address bus consists of four electrically isolated 8-bit bytes. Two of these bytes are used to convey the 16 bits comprising each address, a third byte is used for the parity bits associated with each address, and the fourth byte serves as a spare should any of the other bytes fail. The selection of the appropriate set of three bytes to be used is accomplished by a byte rippler at each address bus interface. The byte rippler is controlled by voted status signals from the CCUs.

5.1.2 DATA BUS

Each 32-bit data word is encoded by appending to it the eight parity bits determined by a shortened cyclic code structure. The cyclic code used is that defined by the generator polynomial, $g(x) = (x^6 + x + 1)(x^2 + x + 1)$.

The data bus consists of six electrically isolated 8-bit bytes. Four of these bytes are used to convey the 32 bits comprising each data word, a fifth byte is used for the parity bits associated with each word, and the sixth byte serves as a spare should any of the other bytes fail. The selection of the appropriate set of five bytes to be used is accomplished by a byte rippler at each data bus interface. The byte ripplers are controlled by voted status signals from the CCUs.

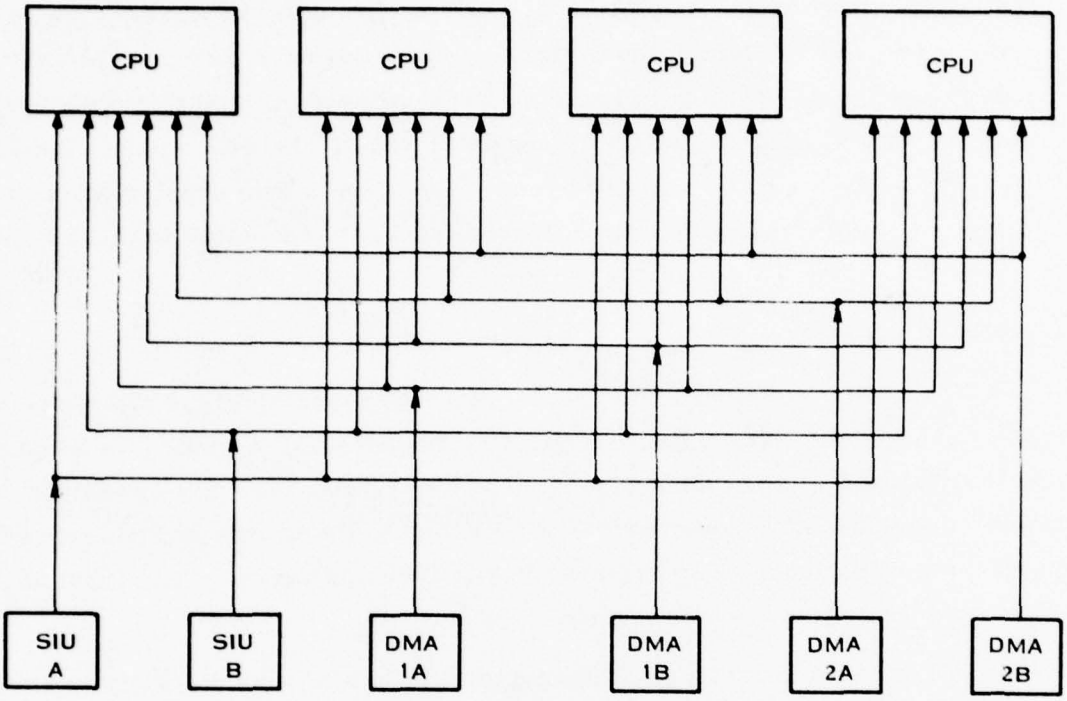
5.1.3 CONTROL BUS

The bus access request and acknowledge lines are configured as shown in Figure 5-1(a) and 5-1(b). Each of the I/O modules has a dedicated bus access request line to, and bus access acknowledge line from, the CPU ensemble. The bus access request and acknowledge lines associated with one I/O module of a redundant pair are electrically isolated in the CPUs from those associated with the other module of the redundant pair.

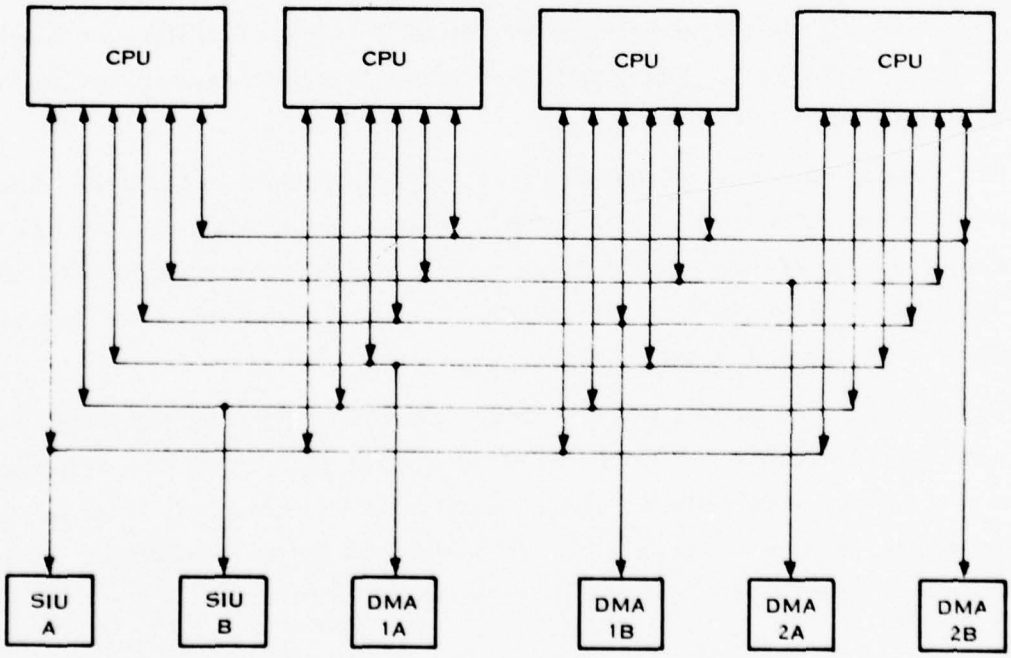
Any I/O module requiring address and data bus access in order to initiate a data transfer to or from a Memory Module raises its bus access request line but does not attempt to effect the data transfer until its bus access acknowledge line is raised. The bus access request line is lowered when the bus access acknowledge is received. The completion of the transfer causes the bus access acknowledge line to be lowered.

The configuration of the data request, acknowledge, read/write, and hard/soft lines is as shown in Figure 5-2. There are three lines for each of these signals; they are connected in a triple-modular-redundant (TMR) configuration. Each of the three lines is electrically isolated from the other two in all modules to which they are connected.

The module controlling the data transfer (the requesting module), once it has access to the buses, sets the read/write line to indicate the direction of the transfer, places the address of the location to be accessed on the address bus, places the data to be transferred (if a write is being performed) on the data bus and raises the request line to initiate the transfer.



(a). Request



(b). Acknowledge

Figure 5-1. Bus Access Lines

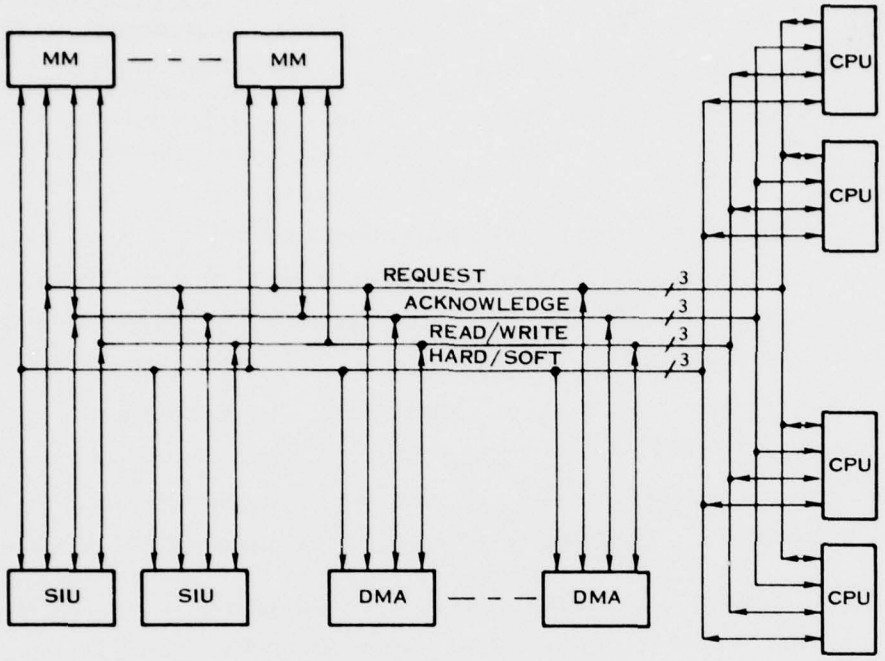


Figure 5-2. Data Request/Acknowledge, Read/Write and Hard/Soft Control Lines

When information is to be transmitted from the requesting module to the addressed module (the read/write line is set to the write state), the addressed module raises the acknowledge line only after it has decoded its address and verified that the address and data are properly encoded. When information is to be transmitted from the addressed module to the requesting module (the read/write line is set to the read state), the addressed module raises the acknowledge line only after it has decoded its address, verified the address code, and placed the requested information on the data bus.

When the requesting module receives the acknowledge, it lowers the request line. The acknowledge line is lowered one system clock period after it is raised (except under certain failure conditions in the memories (cf. Section 5.3.3)).

The BFTSC is able to operate in two addressing modes - the soft address mode and the hard address mode (cf. Section 5.2). The soft address mode is used for all normal computer operations; the hard address mode is used only

by the CPUs for reconfiguration and test purposes. The signal on the hard/soft line indicates the addressing mode; it is generated by the active CPU and received by all the other modules on the bus.

5.1.4 INTERRUPT BUS

The interrupt bus is configured as shown in Figure 5-3. There are two interrupts (end-of-block and general) generated by each I/O (DMA and SIU) module and bused among all four CPU Modules. Each Timing Module generates one real-time interrupt; it is bused in a loop to all CPU and SIU Modules and returned to the Timing Module to be monitored. Interrupts are signalled over these lines by logic level "zero" pulses having a duration of one system clock period. All interrupts associated with one module of a redundant pair are electrically isolated in the CPUs and SIUs from those associated with the other module.

One fault interrupt line emanates from each of the three CCUs to the CPU complex; they are voted at the receivers in each CPU. Fault interrupts are signalled by logic level "one" pulses of duration equal to or greater than one system clock period. Each of the three fault interrupt lines is electrically isolated at the receivers from the other two.

The arithmetic and illegal op code interrupts are both generated and used only within the CPU and hence do not require intermodule signal paths.

5.1.5 STATUS BUS

Figures 5-4 and 5-5 show the configuration of the status lines. The CCU dedicated status lines (Figure 5-4) are shown for one CCU; each of the other two CCUs has an identical set of lines to all of the modules shown. All of the lines associated with one CCU Module are electrically isolated from those associated with the other two CCU Modules in all modules to which they connect. Each non-CCU dedicated status line (or pair of lines for power and timing status) within a group (Figure 5-5) is electrically isolated from all the other lines (or pair of lines) in that group.

There are two power status lines from each Power Module and two timing status lines from each Timing Module; a Power or Timing Module raises both of its status lines when it is in the active state and its outputs are within

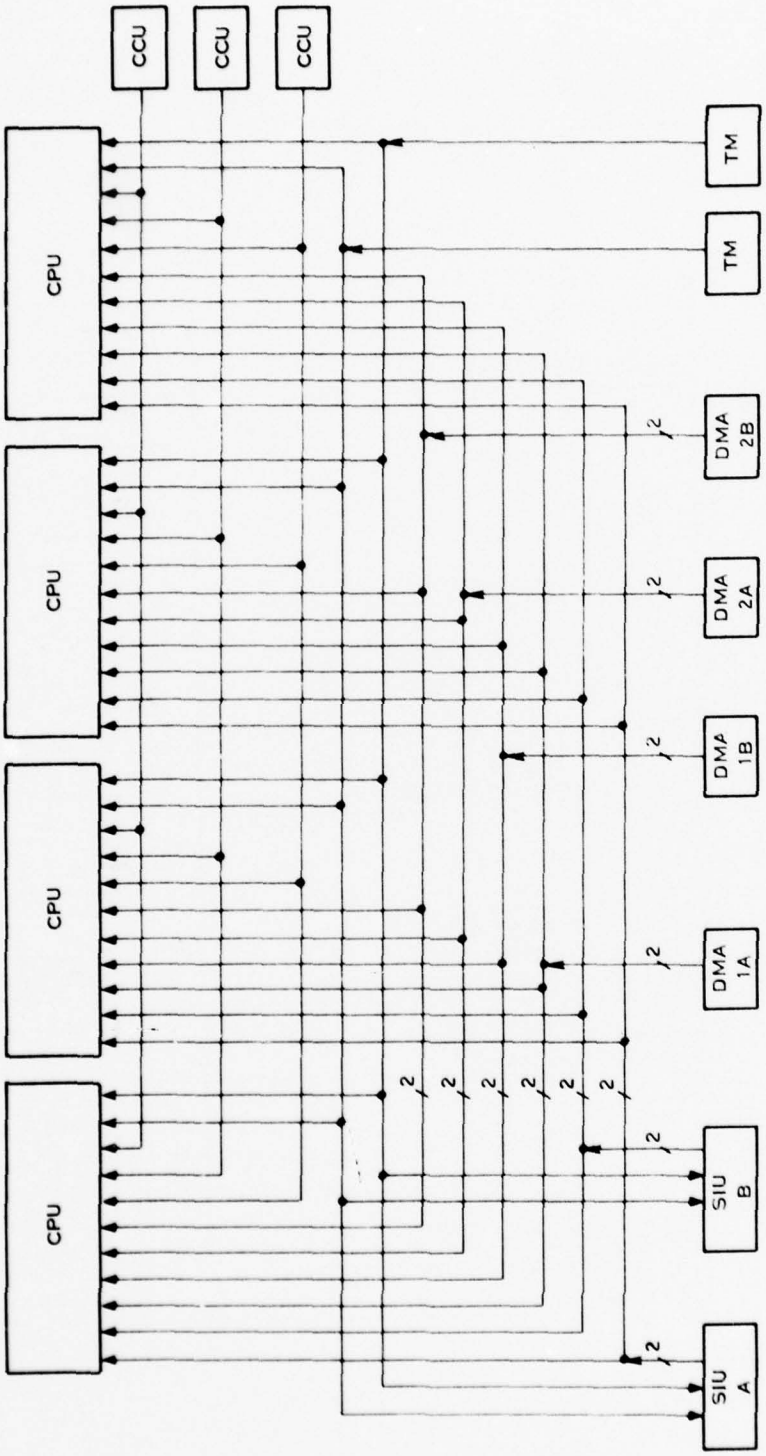


Figure 5-3. Interrupt Bus Configuration

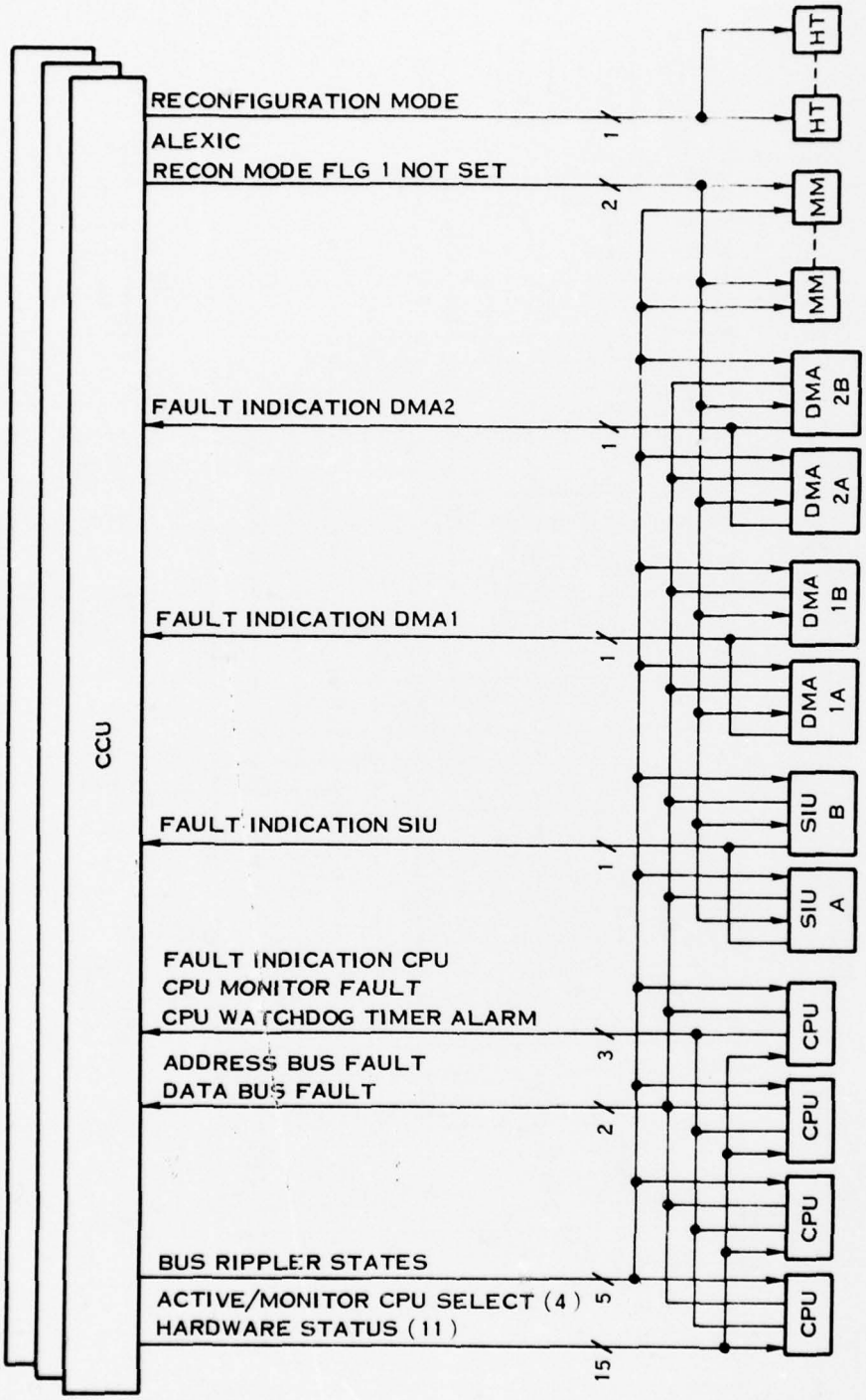


Figure 5-4. CCU Dedicated Status Lines

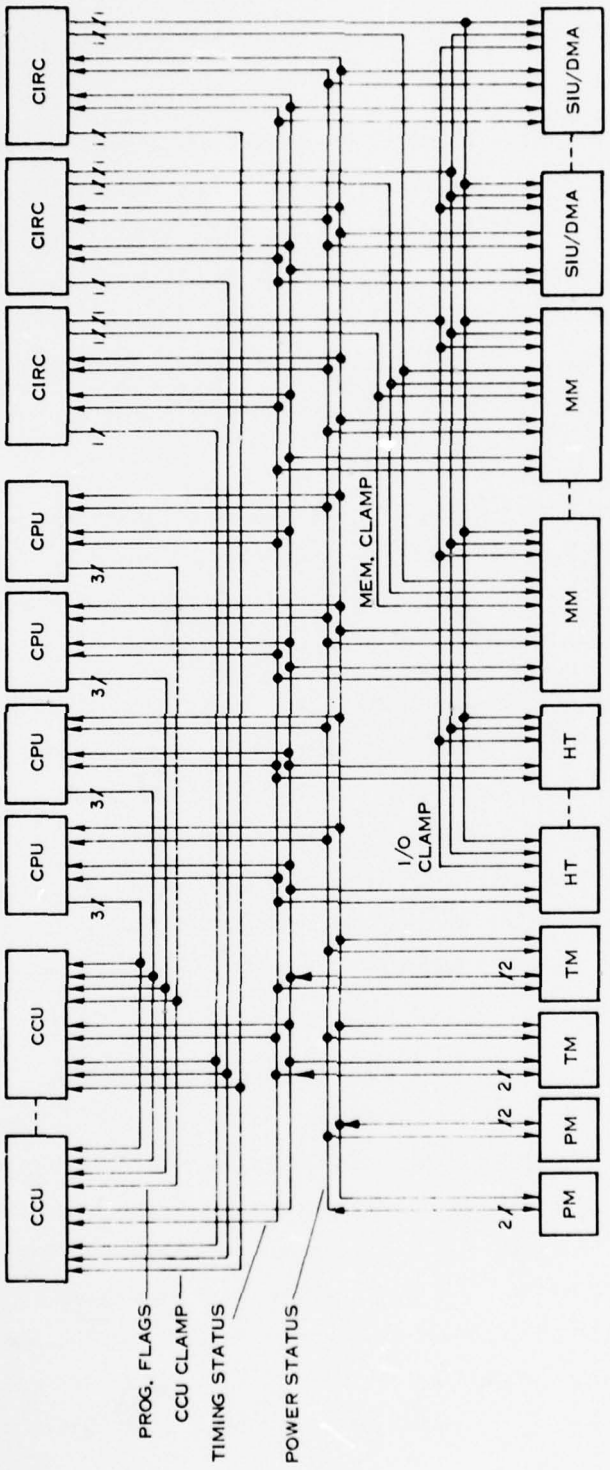


Figure 5-5. Non-CCU Dedicated Status Lines

tolerance. A module uses the outputs from a Power or Timing Module only if both status lines associated with that Power or Timing Module are valid (logic level "one").

There are three clamp lines from each Circumvention Module: a CCU clamp, an I/O clamp, and a Memory clamp. The CCU clamp is set to indicate to the CCU that a radiation event or a power switchover is taking place. The I/O and Memory clamps are raised to indicate power switchover, timing switchover, radiation event, or fault interrupt. The I/O clamp connects to all I/O modules (DMAs and SIUs) and Memory Modules; the Memory clamp connects to all Memory Modules. A clamp is indicated by a logic level "zero" on the clamp lines.

Each CPU sends reconfiguration program flags to the CCU over the program flag lines associated with that CPU as shown in Table 5-2. One of seven possible flags is signalled by the CPU by placing the three bit code for that flag on the status lines for a duration of one system clock period.

Three reconfiguration status signals are generated by each CCU: alexic; reconfiguration mode; and reconfiguration mode, flag 1 not set. When a logic level "zero" pulse of duration equal to or greater than one system clock period occurs on the alexic line, all Memory, DMA, and SIU Modules are powered off. The reconfiguration mode line indicates that the reconfiguration mode has been entered. The reconfiguration mode, flag 1 not set line causes all bus drivers in all Memory, DMA, and SIU Modules to inhibit transmission. The drivers are inhibited as long as this signal is at a logic "zero" level.

Each CCU receives four fault-indication lines, each of which is bused among all sources of that fault indication. The address bus fault and data bus fault lines are bused among all CPU, DMA, and SIU Modules and indicate that an address bus parity fault or data bus parity fault has been detected. A logic level "zero" pulse of duration equal to or greater than one system clock period signals this type fault. The CPU watchdog timer alarm line is bused among all four CPU Modules; it is generated by the active CPU and indicates, with a logic level "one" pulse having a duration of one system clock period, a lack of response to a bus transfer request. The CPU monitor fault line is bused among all CPU

Table 5-2. Reconfiguration Program Flags

Program Flag	Function	Reset by
001	Causes fault interrupt and entry (or reentry) into alexic state.	
010	Causes CPU or bus state to be changed; if the next program flag is 001, CPU state is changed; if it is 010, the address-bus state is changed; and if it is 100, the data-bus state is changed. Causes a fault interrupt.	
011	Returns system to normal mode and resets CCU inputs to the Hardware Status Word.	
100	Sets "Flag 1", allows modules to respond to their addresses, prevents change in state that would otherwise be caused by a CAT III fault and, in alexic mode, by CPU-monitor-detected CAT II faults.	CAT I faults, change in CPU state, entry or reentry into alexic mode
101	Sets "Flag 2" which inhibits state changes and fault interrupts due to CPU-monitor-detected CAT II faults, all CAT III faults, and all CAT IV faults except timing-module switchover signals.	Any fault; program flag 111
110	Sets "Flag 3" which inhibits fault interrupts caused by all CAT III faults and all CAT IV faults except Timing Module switchover signals.	Any fault; program flag 111
111	Resets Flags 2 and 3.	

Modules; it is generated by the monitor CPU and indicates with a logic level "one" pulse having a duration of one system clock period, a discrepancy between the monitor and active CPU Modules.

In order to indicate the source of an address or data bus fault report, four lines are provided to each CPU Module. One line is bused among all four CPUs, one is bused between both SIU Modules, one is bused between both DMA1 Modules, and one is bused between both DMA2 Modules. (The DMA2 Modules are not present in the BFTSC.) A pulse generated on the CPU fault, SIU fault, DMA1 fault, or DMA2 fault line indicates that an address or data bus fault has been detected by the active or monitor CPU, the powered SIU, the powered DMA1, or the powered DMA2 respectively.

Each CCU generates two active CPU select lines and two monitor CPU select lines. When the two-bit code on the active or monitor select lines corresponds to the preassigned code for a particular CPU, that CPU is powered on as the active or monitor CPU.

Each CCU generates two address bus rippler state lines and three data bus rippler state lines. The code on these lines determines which three address bus bytes and which five data bus bytes are active.

The first eleven hardware status bits (cf. Table 5-4) are generated by each CCU; they provide information to the CPU Modules regarding the state of the BFTSC at the time of the last fault. They are bused among all CPU Modules.

5.1.6 TIMING BUS

Two system clock lines, one from each Timing Module, are bused in a loop to all BFTSC modules requiring the system clock and back to the Timing Module in order to be monitored. Each clock line is electrically isolated, at each module, from the other line so that no single fault can affect more than one Timing Module.

Each hardened Timer Module generates one output to and receives one input from the CPU complex. The lines associated with Each Hardened Timer Module are electrically isolated, in all the CPU Modules, from those associated with the other Hardened Timer Modules. When in the count down mode, the

Hardened Timer raises its output line each time one stored RTI period has been counted. The output line is lowered whenever the reset line from the CPU is pulsed (logic level "one" pulse of one system clock period duration). The first pulse on the reset line begins the countdown process.

5.1.7 POWER BUS

Each required voltage is distributed to all the BFTSC modules by two lines, one from each Power Module. Each of these lines is bused in a loop from the Power Module to all other modules requiring it and back to the Power Module to be monitored. Two energy storage lines from each Power Module are bused to all Timing, Hardened Timer, and Circumvention modules; they provide power to those modules during power switchovers. All of the lines associated with one Power Module are electrically isolated, at each BTFSC module, from those lines associated with the other Power Module so that no single failure can affect both Power Modules.

5.2 ADDRESS SPECTRA

In order to facilitate the transmission of configuration status and control information, the BFTSC is able to operate using two addressing modes: a soft address mode, and a hard address mode. The soft address mode is used for all normal operations; the hard address mode is activated only by the CPU (using the store hard [STH] instruction) for reconfiguration and test purposes.

5.2.1 SOFT ADDRESS SPECTRUM

The BFTSC has a 64K word soft address spectrum as defined in Tables 5-3 and 5-4. The first 60K words (0000 - EFFF) are reserved for random access read/write memory. The read/write Memory Module configuration can be altered under program control by changing the soft name (the first four bits of the soft address) assigned to each Memory Module. The soft name defines the set of 4K addresses to which the module will respond within the 60K spectrum.

The next 2K words (F000 - F7FF) of the soft address spectrum are reserved for the reconfiguration ROM which contains the recovery program. The last 2K words of the soft address spectrum (F800 - FFFF) are reserved for I/O addressing and all of the special functions required for the BFTSC. The data bit assignments for all of the special soft addresses are listed in Table 5-4.

Table 5-3. Soft Address Spectrum

Soft Address	Assigned Function
0000-0FFF	Memory Module assigned soft name 0
1000-1FFF	Memory Module assigned soft name 1
2000-2FFF	Memory Module assigned soft name 2
3000-3FFF	Memory Module assigned soft name 3
4000-4FFF	Memory Module assigned soft name 4
5000-5FFF	Memory Module assigned soft name 5
6000-6FFF	Memory Module assigned soft name 6
7000-7FFF	Memory Module assigned soft name 7
8000-8FFF	Memory Module assigned soft name 8
9000-9FFF	Memory Module assigned soft name 9
A000-AFFF	Memory Module assigned soft name A
B000-BFFF	Memory Module assigned soft name B
C000-CFFF	Memory Module assigned soft name C
D000-DFFF	Memory Module assigned soft name D
E000-EFFF	Memory Module assigned soft name E
F000-F3FF	Reconfiguration ROM
F400-F7FF	Not assigned
F800	Hardware Status Word 1
F801	Hardware Status Word 2
F802	Monitor CPU Mask
F803	Peripheral Mask
F804	Interrupt Mask
F805	Downcount Hardened Timer
F806-F808	Not assigned
F809-F80F	Program Flags to CCU
F810-F83F	Not assigned
F840	DMA1 Control/Status Word 0
F841	DMA1 Control/Status Word 1
F842	DMA2 Control/Status Word 0
F843	DMA2 Control/Status Word 1

Table 5-3. Soft Address Spectrum (Continued)

Soft Address	Assigned Function
F844	SIU Control/Status Word 0
F845	SIU Control/Status Word 1
F846-FCAF	Not assigned
FCB0-FCBE	Not assigned
FCBF-FDCF	Not assigned
FDD0-FDDE	New Rippler Register Low Order for Soft Names 0-E
FDDF-FE9F	Not assigned
FEA0-FEAE	Memory Fault Status Register for Soft Names 0-E
FEAF-FFBF	Not assigned
FFC0-FFCE	New Rippler Register High Order for Soft Names 0-E
FFCF-FFFD	Not assigned
FFFE	Control Data Display
FFFF	Control Panel

Table 5-4. Soft Address Word Formats

	Bit	Function
Interrupt Return (0000-0008)	8	Interrupt Disable Status
	9	Divide Check Status
	10	Overflow Status
	11	Illegal Op Code Status
	12	Carry out Status
	13-15	Interrupt Level
	16-31	Return Address
Hardware Status Word 1 (F800) (load)	0-2	Flags 1, 2 and 3
	3	Zero
	4	Alexic
	5-8	Fault Category/Subcategory 1111 = Cat I Circumvention or Power Switchover 1100 = Cat II 1000 = Cat III Address Bus 1001 = Cat III Data Bus 0011 = Cat IV SIU 0001 = Cat IV DMA1 0010 = Cat IV DMA2 0110 = Cat IV CPU WDT 0111 = Cat IV Timing Module Switchover
	9	CPU/BUS State-Change Bit (1 = either CPU or one of the buses has changed state during the current reconfiguration)
	10	CPU Test Bit (0 = Current pair of CPUs has been tested)
	11	CPU Simplex Mode
	12-13	State of Bus Arbiter* 00 = SIU 01 = DMA1 10 = DMA2 11 = CPU
	14	Read/Write control line* (1 = Read)

*These inputs represent the contents of the signals at the time of the fault interrupt.

Table 5-4. Soft Address Word Formats (Continued)

	Bit	Function
	15	Hard/Soft Control Line* (1 = Hard)
	16-31	CPU's most recent address register*
Hardware Status Word 2 (F801) (Load)	0-2	Data-Bus State Bit 0 1 2 — Rippled Out Byte 0 0 0 — 5 0 0 1 — 4 0 1 0 — 3 0 1 1 — 2 1 0 0 — 1 1 0 1 — 0
	3-4	Address Bus State Bit 3 4 — Rippled Out Byte 0 0 — 3 0 1 — 2 1 0 — 1 1 1 — 0
	5-8	CPU State Bit 5 6 7 8 0 0 X X — CPU 1 - Active 0 1 X X — CPU 2 - Active 1 0 X X — CPU 3 - Active 1 1 X X — CPU 4 - Active X X 0 0 — CPU 1 - Monitor X X 0 1 — CPU 2 - Monitor X X 1 0 — CPU 3 - Monitor X X 1 1 — CPU 4 - Monitor
	9	Power Module State 0 = Power Module A Active
	10	Timing Module State 0 = Timing Module A Active
	11-13	Hardened Timer Outputs

*These inputs represent the contents of the signals at the time of the fault interrupt.

Table 5-4. Soft Address Word Formats (Continued)

	Bit	Function
Monitor CPU Mask (F802) (Store)	24	Downcount Hardened Timer* (1 = Mask)
	25	BA DMA1 ACK* (1 = Mask)
	26	BA DMA2 ACK* (1 = Mask)
	27	BA SIU ACK* (1 = Mask)
	28	Read/Write* (1 = Mask)
	29	Hard/Soft* (1 = Mask)
	30	Request* (1 = Mask)
	31	Watchdog Timer* (1 = Mask)
Peripheral Mask (F803) (Store)	24	Clear Interrupts (1 = Clear)
	25	Request (1 - Mask)
	26	DMA1 A (1 - Mask)
	27	DMA1 B (1 - Mask)
	28	DMA2 A (1 - Mask)
	29	DMA2 B (1 - Mask)
	30	SIU A (1 - Mask)
	31	SIU B (1 - Mask)
Interrupt Mask (F804) (Store)	24	Arithmetic Fault (1 = Mask)
	25	RTI (1 = Mask)
	26	SIU (General) (1 = Mask)
	27	DMA1 (General) (1 = Mask)
	28	DMA2 (General) (1 = Mask)
	29	SIU (EOB) (1 = Mask)
	30	DMA1 (EOB) (1 = Mask)
	31	DMA2 (EOB) (1 = Mask)
Downcount Hardened Timer (F805) (Store)	0-31	Don't Care
Program Flags to CCU (F809-F80F) (Store)	0-31	Don't Care (F809 = Flag 1, F80F = Flag 7)

*Monitor CPU comparator input inverted.

Table 5-4. Soft Address Word Formats (Continued)

	Bit				Function
DMA1, 2 Control Word 0 (F840, F842) (Store)				0	Start
				1	Stop
				2	Data Receive
				3	Data Transmit
				4-15	Stop Address
				16-31	Start Address
DMA1, 2 Status Word 0 (F840, F842) (Load)				0	Any Failure
				1	Address-Bus Fault
				2	Data-Bus Fault
				3	Control Word Error
				4	WDT-BA
				5	Internal Failure
				6	Busy
				7	User Request
				8	Command Done
				9	User Busy
				10-15	Zeros
				16-31	Most Recent Address Register
DMA1, 2 Control Word 1 (F841, F843) (Store)	0	1	2	3	
	1	0	0	0	Comparator Test Disarm
	1	1	0	0	Arm MSB Comparator Test
	1	0	1	0	Arm LSB Comparator Test
	1	0	0	1	Arm Error Code Comparator Test
	0	X	X	X	User Command (with bits 1-31)
DMA1, 2 Status Word 1 (F841, F843) (Load)				0-31	User Request

Table 5-4. Soft Address Word Formats (Continued)

	Bit	Function
SIU Control Word 0 (F844) (Store)	0	Immediate Block
	1 2 3	
	0 0 0	Normal Block
	0 0 1	Switch Receive Bus
	0 1 0	Invert last Error Code bit
	0 1 1	Zero Error Code bits
	1 0 0	Inhibit Error Code bits
	1 0 1	Invalid last Error Code bit
	1 1 0	Invalidate first Sync bit
	1 1 1	Normal Block
	4-15	Stop Address
	16-31	Start Address
SIU Status Word 0 (F844) (Load)	Bit	
	0	Any Failure
	1	Address-Bus Fault
	2	Data-Bus Fault
	3	Invalid Waveform on Serial Bus
	4	WDT-BA
	5	Internal Failure
	6	Busy
	7	Last Memory Access Fetch
	8	Receiving Bus A
	9	Bad Parity on Serial Bus
	10-15	DIU Priority Interrupt
	16-31	Most Recent Fetch Address
SIU Control Word 1 (F845) (Store)	0 1 2 3	
	1 0 0 0	Disarm Comparator Tests
	1 1 0 0	Arm MSB Comparator Test
	1 0 1 0	Arm LSB Comparator Test
	1 0 0 1	Arm Error Code Comparator Test
	0 X X X	Halt
SIU Status Word 1 (F845) (Load)	0-15	Same as Status Word 0
	16-31	Most Recent Store Address

Table 5-4. Soft Address Word Formats (Continued)

	Bit	Function
Memory Fault Status Register for Soft Names 0-E (FEA0-FEAE) (Load)	0	Fault (One of following bits set)
	01-12	All Zeros
	13*	Armed Control Functions are Armed
	14*	Error Correction is Armed
	15*	Fault Override is Armed
	16	Block Fault
	17	Soft Name Compare Fault
	18	Write Protect Violation
	19	Address-Bus Fault
	20	Address Decode Fault
	21	Channel Select Decode Fault
	22	X/Y Switch Fault
	23	IW Missing Fault
	24	Timing/ID Missing Fault
	25	Data Bus Fault
	26	Data Code from Interface Fault
	27	Memory Data Fault
	28	Rippler Overflow Fault
	29	Syndrome Fault
	30	Incorrect Refresh Request
	31	Zero
New Rippler Register Low Order for softnames 0-E (FDD0-FDDE) (L/S)	0-22	New Rippler Register Bits 0-22 (Load Only)
	23-31	New Rippler Register Bits 32-40
New Rippler Register High order for softnames 0-E (FFC0-FFCE) (L/S)	0-31	New Rippler Register Bits 0-31

*Armed Flags Not Included in Fault (Bit 0)

Except for the store to hard address (STH) instruction, all instructions use the soft address spectrum. Addressing a nonexistent or unpowered module in the soft address mode causes a fault interrupt due to a lack of response from the addressed device.

5.2.2 HARD ADDRESS SPECTRUM

Each Memory, DMA, and SIU Module has a unique preassigned five-bit hard address (listed in Table 5-5) to which it responds when this address appears as the most significant bits (bits 0-4) on the address bus and data bus. The 11 bits following the hard address define the function performed by the addressed module. The hard address functions are defined in Table 5-6. The definitions of mnemonics for the hard address functions are given in Table 5-7. Simultaneous recognition of its hard address on both the address and data buses in the hard address mode is required to perform any of the hard address functions except power-off. Recognition of its hard address combined with three zeros in bits 5-7 on either the address or data bus in the hard address mode is sufficient to turn off power to any module.

The hard address mode is accessed only by the CPUs using the store to hard address (STH) instruction (cf. Appendix A). Hard addressing does not require an acknowledge from the addressed device; therefore, addressing a non-existent module in the hard address mode does not cause a fault interrupt.

Table 5-5. Hard Address Assignments

Hard Address	Module
0	Not Assigned
1-24	Memory
25	SIU A
26	SIU B
27	DMA1 A
28	DMA1 B
29	DMA2 A
30	DMA2 B
31	Not Assigned

Table 5-6. Hard Address Functions

Hard Address	Address Bits 5	6	7	8	9	10	11	12	13	14	15
Function											
PWRON	1	1	1	-	-	-	-	-	-	-	-
PWROFF	0	0	0	-	-	-	-	-	-	-	-
Memory Function											
LDSFTN	0	0	1	A_0	A_1	A_2	A_3	A_0	A_1	A_2	A_3
LDWP	0	1	0	-	-	-	-	A	B	C	D
ARMED CONTROL FUNCTIONS											
IGW } Both = Dormant	0	1	1	0	-	-	-	-	-	-	-
IGR } Both = Dormant	0	1	1	-	0	-	-	-	-	-	-
STNEWST	0	1	1	-	-	1	-	-	-	-	-
EZPARA	0	1	1	-	-	-	1	-	-	-	-
TST ONE	0	1	1	-	-	-	-	1	-	-	-
TST ZERO	0	1	1	-	-	-	-	-	1	-	-
STECORR	0	1	1	-	-	-	-	-	-	1	-
STRFRSH	0	1	1	-	-	-	-	-	-	-	1
AMRESET	0	1	1	1	1	0	0	0	0	0	0
UPDORR	1	0	0	-	-	-	-	-	-	-	-
FAULT OVERRIDE											
EBADPARD	1	0	1	-	1	-	-	-	-	-	-
FANLGFO	1	0	1	-	-	1	-	-	-	-	-
ENWAFO	1	0	1	-	-	-	1	-	-	-	-
FMRESET	1	0	1	0	0	0	0	-	-	-	-
*ARM TEST MEMORY MONITORS (Self Resetting)											
IWMIS	1	1	0	1	-	-	-	-	-	-	-
CSSU	1	1	0	-	1	-	-	-	-	-	-
ITDCLR	1	1	0	-	-	-	1	0	0	0	0
ITDCLK	1	1	0	-	-	-	1	0	0	1	0
ITID	1	1	0	-	-	-	1	0	1	0	0
ITSTCH	1	1	0	-	-	-	1	0	1	1	0
XAON	1	1	0	-	-	-	1	1	0	0	0
XBON	1	1	0	-	-	-	1	1	0	1	0
YAON	1	1	0	-	-	-	1	1	1	0	0
YBON	1	1	0	-	-	-	1	1	1	1	0
XAOFF	1	1	0	-	-	-	1	1	0	0	1
XBOFF	1	1	0	-	-	-	1	1	0	1	1
YAOFF	1	1	0	-	-	-	1	1	1	0	1
YBOFF	1	1	0	-	-	-	1	1	1	1	1
TMRESET	1	1	0	0	0	0	0	0	0	0	0

*Causes an extended acknowledge on the next soft name access.

Table 5-7. Hard Address Function Mnemonic Definition

Address	Definition
PWRON	Power On
PWROFF	Power Off
LDSFTN	Load Soft Name
LDWP	Load Write Protect
IGW*	Ignore Write
IGR*	Ignore Read
STNEWST	Set New Status
EZPARA	Enable Zeros Parity on Address
TST ONE	Test Ones
TST ZERO	Test Zeros
STECORR	Set Error Correction
STRFRSH	Set Refresh
AMRESET	Master Reset Armed Control Functions
UPDORR	Update Old Rippler Register
FMRESET	Master Reset Fault Override Functions
EBADPARD	Enable Bad Parity on Data
EANLGFO	Enable Analog Fault Override
ENWAFO	Enable Name Compare, Write Protect, Address Decode Fault Override
IWMIS	Word Current Missing
CSSU	Channel Select Fault
ITDCLR	Inhibit Timing Pulse Data Clear
ITDCLK	Inhibit Timing Pulse Data Clock
ITID	Inhibit Timing Pulse Digit Current
ITSTCH	Inhibit Timing Pulse Stack Charge
XAON, XBON	X Switch Word Select A/B on Fault
YAON, YBON	Y Switch Word Select A/B on Fault
XAOFF, XBOFF	X Switch Word Select A/B Off Fault
YAOFF, YBOFF	Y Switch Word Select A/B Off Fault
TMRESET	Master Reset Test Memory Monitor Functions

*Dormant Mode is set by setting both IGW and IGR.

5.3 SYSTEM MODULES

5.3.1 CENTRAL PROCESSING UNIT (CPU)

The BFTSC contains four Central Processing Units (CPUs), two of which are always powered during normal BFTSC operation with one denoted by the CCU as the active CPU and the other as the monitor CPU. The monitor compares all control, address, data, and watchdog-timer outputs from the active CPU with its own internally generated signals. The monitor informs the CCU of any discrepancy between the two CPUs. The monitor does not drive any outputs other than those dedicated to the CCU.

Figure 5-6 is a block diagram of the CPU. It is implemented in five general sections: the interface, the register arrays and arithmetic logic unit (ALU) section, the priority interrupt network, the control section, and the fault monitoring and configuration status section. The interface section, which is always powered, contains address-bus and data-bus ripplers, coders, and comparators. It serves as the interface between the CPU and all internal BFTSC buses. It votes on the logic levels of the "active CPU" and "monitor CPU" status lines emanating from each of the three CCUs and generates power control signals in accordance with this information to power on the remaining sections of the CPU. These power control signals are raised if and only if either the active CPU or the monitor CPU status lines indicate the two-bit pattern corresponding to the two-bit designation preassigned to that CPU.

The register arrays and ALU section contain the arithmetic logic unit and two sets of 8-word by 32-bit register arrays. The eight registers in one of these sets serve as general purpose addressable registers. The second set of registers includes the extension register, program counter, memory address and memory data registers, and four working registers. The ALU is capable of performing all of the arithmetic and logical operations implied by the specified instruction set (cf. Appendix A).

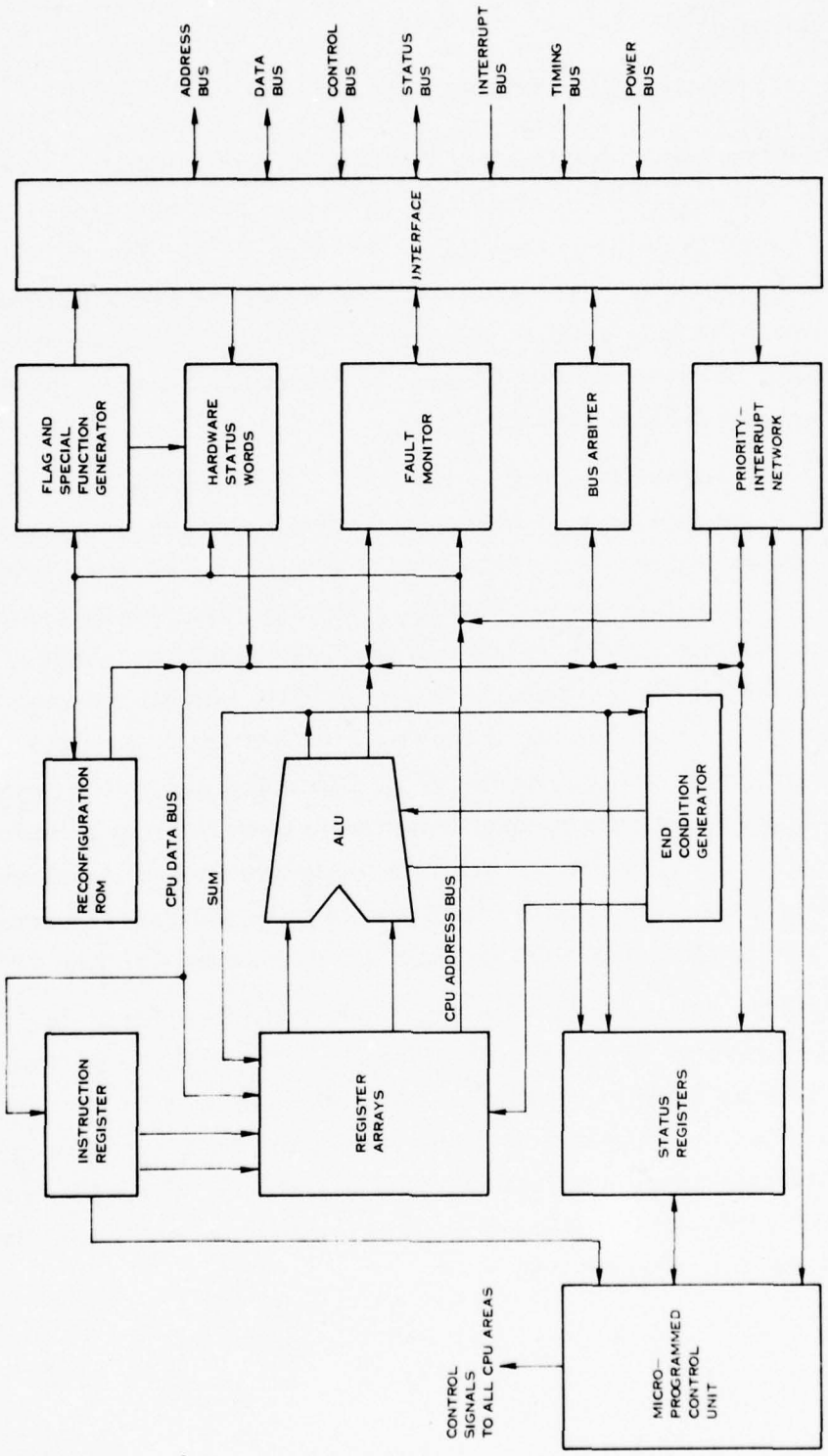


Figure 5-6. CPU Functional Block Diagram

The priority interrupt network (PIN) recognizes the ten levels of interrupts identified as follows (listed in order of decreasing priority):

Fault
Illegal Op Code
Arithmetic
Real-Time Interrupt
SIU (General)
DMA1 (General)
DMA2 (General)
SIU (End of Block)
DMA1 (End of Block)
DMA2 (End of Block)

The eight lower priority interrupts originate as pulses from the requesting devices and are stored in the PIN in latched flip-flops (one flip-flop is provided for each level). The eight lower order interrupts are individually maskable under program control. If the mask for a given interrupt is not set when the interrupt occurs, or when the mask is reset following the occurrence of the interrupt, the request is processed by the PIN if and only if it has higher priority than any other pending unmasked interrupt. The PIN causes the program to branch to an associated interrupt service routine. A return address and appropriate status information (ALU overflow, carryout and divide-check status, interrupt enable/disable status, illegal opcode status and the highest priority pending interrupt) are simultaneously stored into preassigned locations. Servicing any interrupt causes the lower priority interrupts to be disabled, resets the request flip-flop for the level being serviced, and sets an "in-process" flip-flop for that level. It is possible to enable and disable the interrupt network and to reset the "in-process" flip-flop for any given level under program control.

All interrupts except the fault interrupt and the illegal op code interrupt are honored during the instruction fetch cycle. Fault and illegal op code interrupts are honored immediately (at the next microinstruction). These interrupts are not maskable. A fault interrupt resets all in-process and request flip-flops and is serviced even if a previous fault interrupt is still in process.

The control section (cf. Figure 5-6) contains the bus arbiter, the microprogrammed control unit, the instruction register, the state address and status registers, and the end-condition generator.

The bus arbiter (BA) exercises administrative control over access to the address, data, and control buses. The BA resolves conflicting bus access requests by assigning priority in a prescribed cyclic sequence. Requests from the I/O Modules (SIUs and DMAs) are individually maskable by loading the appropriate mask into the peripheral mask register (cf. Table 5-4).

The microprogrammed control unit defines the microinstruction control state for the CPU by sequencing through its control state decode ROM. The specific ROM sequence is determined by the op code and address mode fields of the instruction (stored in the instruction register) and by the branch conditions indicated in the state address and status registers. The next control state is generated in this latter unit from the current control state and all relevant CPU control and data variables. The end-condition generator generates control signals for the ALU and the two register arrays in accordance with the current microinstruction control state.

The fault monitoring and configuration status section includes the reconfiguration ROM, the flag and special function generator, the hardware status word registers, the monitor mode comparators, the watchdog timer, and the fault generator.

The reconfiguration ROM contains all program routines needed to reconfigure an operational BFTSC system following a fault and a bootstrap loader for entering programs into main memory. The 1024 word by 32-bit ROM is accessable in the soft address mode using the assigned addresses listed in Table 5-3.

The flag and special function generator (FSFG) is also addressable in the soft address mode and responds to the (hexadecimal) addresses F800 through F805 and F809 through F80F (cf. Table 5-3). The FSFG responds to addresses F800 through F801 by gating the contents of hardware status word registers 1 and 2, respectively, onto the internal CPU data bus. The composition of these status words is as defined in Table 5-4. Addresses F802, F803, and F804

cause the FSFG to gate the appropriate portion (cf. Table 5-4) of the word on the data bus into the monitor CPU mask register, the peripheral mask register, and the interrupt mask register, respectively. Address F805 causes the FSFG to send a pulse to the hardened timer indicating that the program is ready to read the elapsed time recorded by that device. Addresses F809 through F80F set the program flag register, which is monitored by the CCU and used to coordinate reconfiguration and testing procedures, to the state specified by the last three address bits. This state is reset to 000 after one system clock cycle.

The watchdog timer measures the time interval between the instant a module (CPU, DMA, SIU) is given access to the bus system by the bus arbiter and the instant this access is relinquished. The fault generator sends a watchdog-timer alarm of one system clock cycle duration to the CCU should the watchdog-timer measure a time interval exceeding 10 clock cycles.

The monitor mode comparators (MMC) are used only when the CPU is in the monitor mode. When this is the case, the MMC provides a means for comparing all data, address, and control signal outputs, and the watchdog timer alarm, generated by the active CPU with the corresponding signals generated (but not transmitted externally) by the monitor CPU. The fault generator sends a CPU fault signal of one clock cycle duration to the CCU should the MMC detect any discrepancy. The MMC contains a monitor mask register which can be used to simulate fault conditions and hence test the MMC's monitoring capability. If a bit in the monitor mask register is set to a "one" (cf. Table 5-4), the corresponding monitor CPU input to the MMC is complemented. The monitor mask register is cleared automatically whenever the MMC detects a discrepancy.

5.3.2 CONFIGURATION CONTROL UNIT (CCU)

The Configuration Control Unit (CCU) (cf. Figure 5-7) consists of three general sections: a CPU/bus state sequencer, a watchdog timer, and a fault interrupt and program flag decoder.

The CPU/bus state sequencer section, consisting of the CPU/bus state change logic and the CPU, address bus, and data bus state sequencers (cf. Figure 5-7), determines and broadcasts to the rest of the system the identities of the active and monitor CPUs, the three active address-bus bytes, and the five active data-bus bytes. If an address-bus state change is indicated, and all four

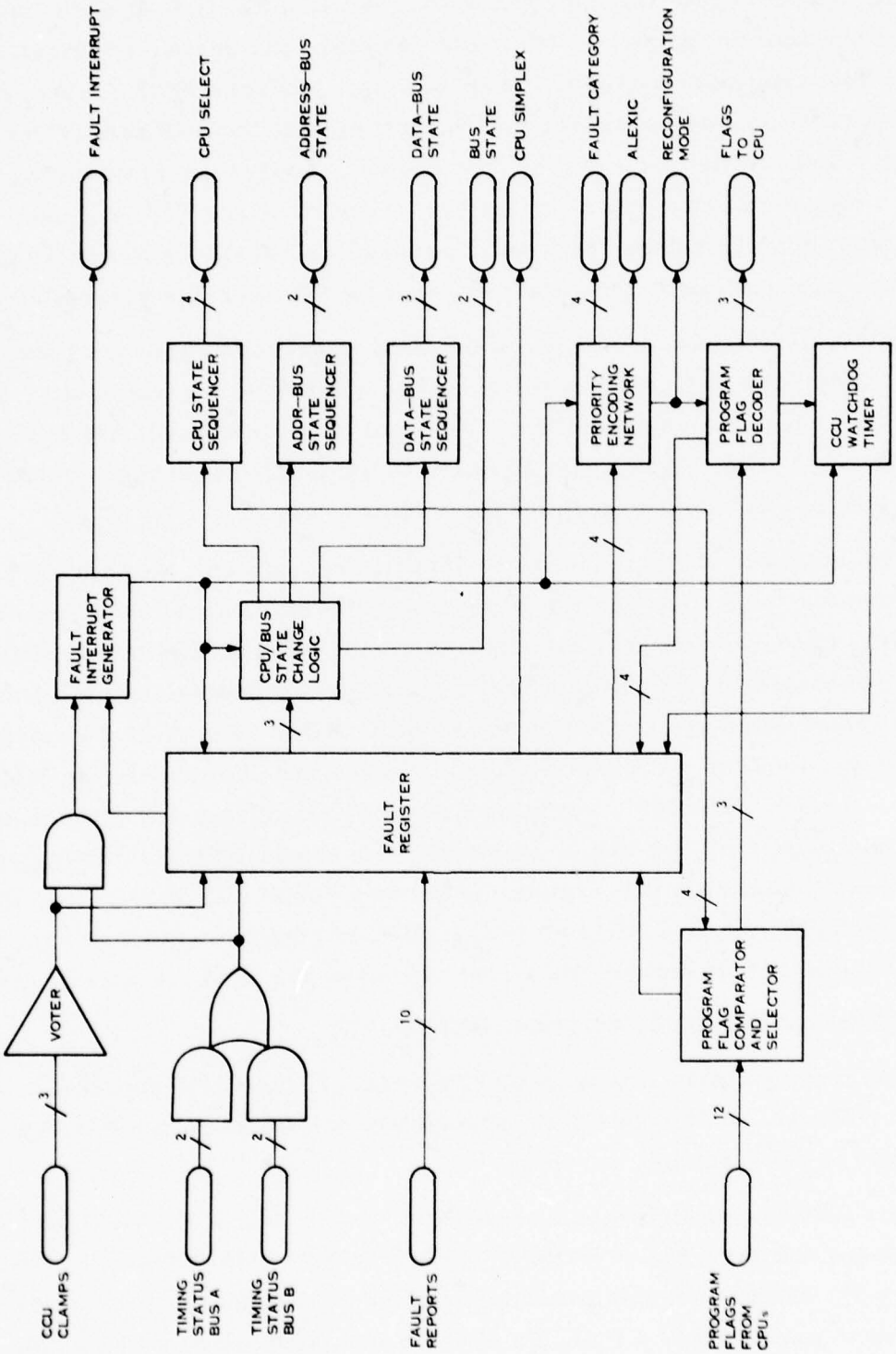


Figure 5-7. CCU Functional Block Diagram

address bus states have already been tried during the current reconfiguration mode, a CPU state change is effected instead. Similarly, if a data-bus state change is indicated and all six data bus states have already been tried in the current reconfiguration mode, the CPU state is instead changed. If six CPU states have been tried during the current reconfiguration mode, the CCU signals the entry (or reentry) into the alexic state (see below) with the next CPU state change. The CPU state sequencer sequences CPU states in such a way that no CPU is powered in more than two consecutive states. If a given CPU state (i.e., a specific pair of powered CPUs) is ever repeated during any continuously powered period, the active and monitor role assignments are reversed from those assignments used previously. The CPU/bus state sequencer generates bits 9 and 10 of HSW1 (cf. Table 5-4).

The watchdog timer measures the time between a fault interrupt and the first setting of a program flag (cf. Table 5-2) and generates an alarm if the time exceeds 8192 clock pulses.

The fault interrupt and program flag decoder section includes a fault register, a fault interrupt generator, a priority encoding network, a program flag decoder, and a program flag comparator.

The fault register contains thirteen flip-flops, each of which is set uniquely by one of the following signals: CCU clamp (when at least two of the Circumvention Units indicate a CCU clamp condition); Timing Module fault (when one or both of the status lines associated with the active timing module show an out-of-tolerance condition); CPU watchdog timer alarm; CPU fault (from the monitor mode comparator); CCU watchdog timer alarm (from the CCU's own watchdog timer); CPU interface fault (address or data bus code parity violation detected by the CPU interface); SIU interface fault; DMA1 interface fault; DMA2 interface fault; data bus fault (detected by a CPU, SIU, or DMA interface); address bus fault (detected by a CPU, SIU, or DMA interface); simplex mode ground override; and state-change ground override.

The latter two signals originate externally to the BFTSC and provide a means for external control of its configuration. The simplex mode ground override signal sets (or, if previously set, resets) a flip-flop in the fault register

which forces the BFTSC into (or out of) the simplex mode. In the simplex mode, the CCU inhibits certain fault interrupts that would otherwise be generated (see below) and changes the CPU state only when the state-change ground override flip-flop is set. The state of the simplex mode flip-flop constitutes bit 11 of HSW1 (cf. Table 5-4).

The fault interrupt generator generates fault interrupts to the CPU in response to the fault listed in Table 5-8. If faults belonging to two or more categories occur simultaneously, the CCU response is determined by the highest-priority fault category or subcategory. (The fault categories and subcategories in Table 5-8 are listed in descending order of priority.) The fault interrupt is kept raised for at least 2048 system clock cycles for all faults resulting in a change in CPU state and for at least two clock cycles for all other faults. When the fault interrupt is lowered, the fault register is cleared.

The priority encoding network encodes the state of the fault register and generates bits 5 through 8 of Hardware Status Word 1 (cf. Table 5-4) indicating the highest priority fault currently reported. The priority encoding network also places the BFTSC system in one of two reconfiguration states. If it announces the alexic state, the power to all Memory and I/O Modules is automatically turned off. The power to any module is turned back on only if that module is appropriately addressed in the hard address mode. If the priority encoding network does not announce the alexic state, all active units remain powered, but no Memory or I/O Module responds to any address on the address bus until Flag 1 has been set by the reconfiguration program (cf. Table 5-2). The priority encoding network selects the alexic reconfiguration state following all CAT I faults and when the appropriate reconfiguration program flag is set (cf. Table 5-2). The priority encoding network generates bit 4 of HSW1 (cf. Table 5-4) making this bit a "one" if the alexic state has ever been entered during the current reconfiguration mode.

The program flag decoder decodes the state of the active CPU's program flag register in accordance with the program flag definitions summarized in Table 5-2 and generates the appropriate fault masks, HSW1 inputs (bits 0-2; cf. Table 5-4), and CPU or bus state change signals. Any disagreement between

Table 5-8. Fault Categories

Fault Category	Subcategory	Caused By	Fault Interrupt and State Change Inhibited By:
I	-	Power on (following an interruption), end of circumvention, or enter or reenter alexic	
II	-	1. CPU-monitor fault signal in the absence of a CAT III fault.	Flag 2; Simplex Mode Flag 1 (when in alexic state)*
		2. Disparity between reconfiguration program flags from active and monitor CPUs.	Simplex Mode
		3. Improper sequence of reconfiguration program flags.	Simplex Mode
		4. Address or data bus fault signal from a CPU bus monitor (in the absence of corroboration from some other monitor of the same bus) when the system is in the normal (nonreconfiguration) mode.	Flags 2 and 3; Simplex Mode
		5. Address or data bus fault detected by more than one bus monitor but not detected by the CPU bus monitor.	Flags 2 and 3; Simplex Mode
		6. CCU watchdog timer alarm	Simplex Mode
III	Multiple Address Bus	1. Fault signals from more than one address-bus monitor when in the normal mode (except when condition 5, above holds).	Flags 2 and 3; Flag 1*; Normal Mode*

*Inhibits state change only

Table 5-8. Fault Categories (Continued)

Fault Category	Subcategory	Caused By	Fault Interrupt and State Change Inhibited By:
III	Multiple Address Bus	2. A fault signal from the CPU address-bus monitor when in the reconfiguration mode.	Flags 2 and 3; Flag 1*; Normal Mode*
	Multiple Data Bus	1. Fault signals from more than one data-bus monitor when in the normal mode.	Flags 2 and 3; Flag 1*; Normal Mode*
		2. A fault signal from the CPU data-bus monitor when in the reconfiguration mode.	Flags 2 and 3; Flag 1*; Normal Mode*
IV	CPU Watchdog Timer	CPU Watchdog Timer Alarm	Flags 2 and 3
	SIU	SIU address or data bus fault signal (in the absence of corroboration from other bus monitors).	Flags 2 and 3
	DMA1	DMA1 address or data bus fault signal (in the absence of corroboration from other bus monitors).	Flags 2 and 3
	DMA2	DMA2 address or data bus fault signal (in the absence of corroboration from other bus monitors).	Flags 2 and 3
	Timing Module	Timing Module switchover	

*Inhibits state change only

the active and monitor CPU program flag registers is detected by the program flag comparator and used to set the CPU fault flip-flop in the fault register. Any attempt to set Flag 1, Flag 2, or Flag 3 (Program Flags 100, 101, or 110) while either Flag 2 or Flag 3 is still in the set state also causes the CPU fault flip-flop to be set.

5.3.3 MEMORY MODULE (MM)

The main memory in the BFTSC is composed of three Memory Modules. The design permits expansion to fifteen active Memory Modules and up to 24 total Memory Modules. In the BFTSC, two Memory Modules are active and one is a spare. One of the Memory Modules is implemented with non-volatile, non-destructive read-out (NDRO), 2-mil plated wire in order to demonstrate the chosen technology for the FTSC. The other Memory Modules are implemented with semiconductor memory, to reduce cost and weight. They are dynamic NMOS with internal refresh so that special control or timing is not required at the module interface. The semiconductor Memory Modules are powered from separate power lines in order to simulate non-volatility. Four dummy Memory Module loads are provided to simulate additional Memory Modules on the data, address, and control buses. Capability is provided to replace the dummy memories with additional (external) Memory Modules.

Figure 5-8 illustrates the Memory Module Functional Block Diagram; details of the Plated-Wire Memory Storage Section and the Semiconductor Memory Storage Section are shown in Figures 5-9 and 5-10 respectively. Functionally, the Memory Modules perform in the following manner.

Upon receipt of a request in the soft address mode, each Memory Module compares its internally stored soft module address with the four most significant bits of the soft address. If a match is obtained, the Memory Module performs the function defined by the address and the read/write control line. The acknowledge is sent when the operation is complete and, in the case of a read, when the data is on the bus. A fault detected prior to acknowledge inhibits the acknowledge. A fault detected during the acknowledge lengthens the acknowledge enough to cause a CPU watchdog timer alarm. In all cases, a detected fault sets an appropriate bit in the Memory Module's fault status register (cf. Table 5-4).

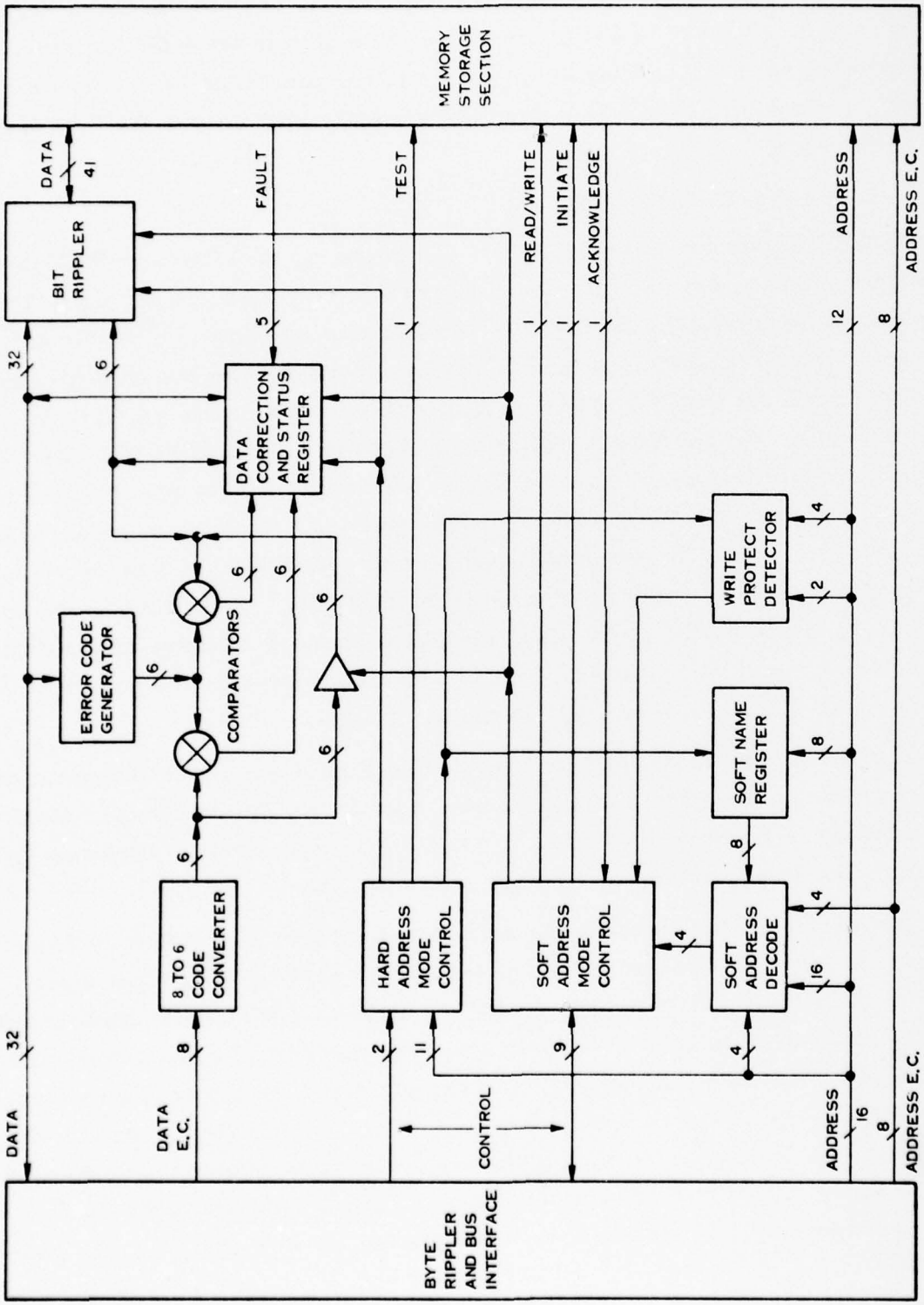


Figure 5-8. Memory Module Functional Block Diagram

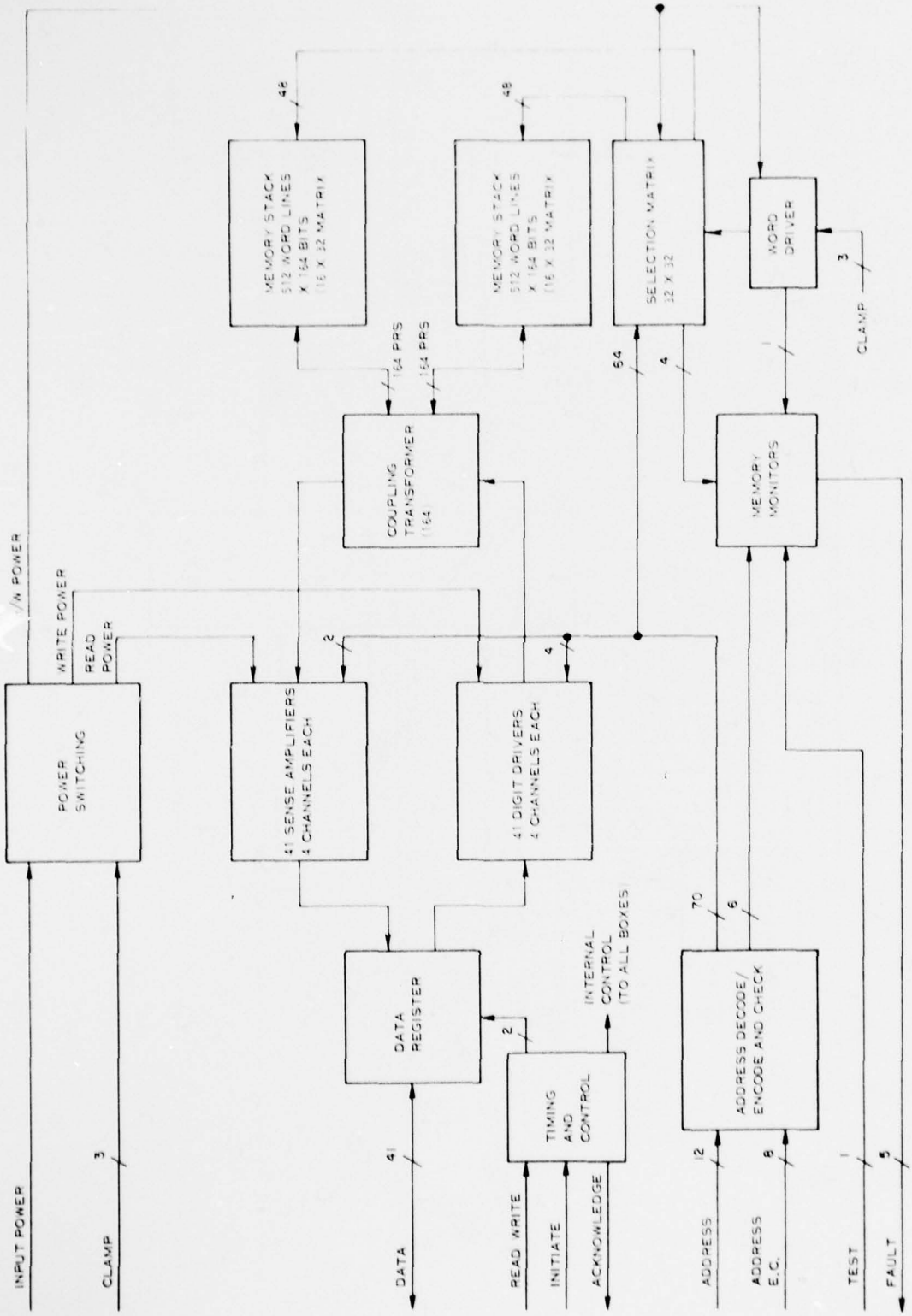


Figure 5-9. Plated Wire Memory Storage Section Block Diagram

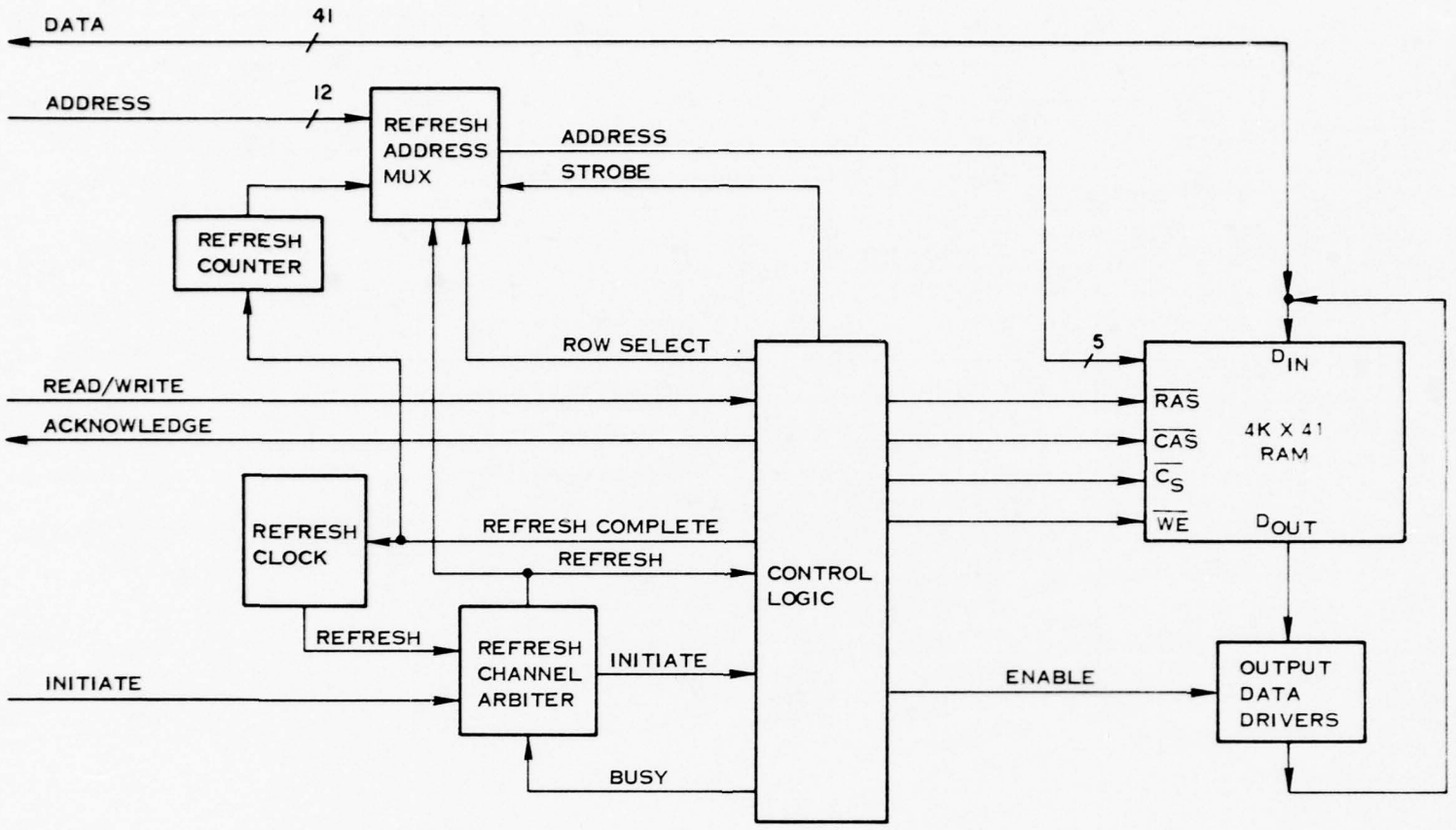


Figure 5-10. Semiconductor Memory Storage Section Block Diagram

The validity of all addresses is checked using the address bus error-detecting code (cf. Paragraph 5.1.1). An invalid address inhibits response and sets a fault status bit in all modules. When write operations are requested, the fifth and sixth bits of the address are compared to the write-protect address bits stored in a Memory Module register. Invalid write requests, i.e., into a write-protected area, inhibit response and set a fault status bit. Address decode circuitry is protected by code testing and selection circuitry is protected by monitoring to prevent false address selection. The Memory Module has monitors to assure that it actually does write when so instructed.

The validity of all received data is checked using the data bus error-detecting code (cf. Paragraph 5.1.2). The eight parity bits associated with this code are then converted to six bits capable of providing single error correction; these six bits are retained with the data throughout all subsequent memory operations. Invalid data causes the memory either to inhibit or to lengthen its acknowledge (depending upon when the fault is detected) and to set the appropriate bit in its fault status register.

The bit rippler contains two rippler state registers, the old rippler register which normally determines the rippler state, and the new rippler register which can be designated the rippler state controller by addressing the Memory Module with the appropriate hard address (cf. Table 5-6). It is possible to read or write into the new rippler register under program control and to force the rippler into either of two test modes. During "test zeros" mode, all "zeros" may be written into any location. A bit in the new rippler register is set to a "one," regardless of its previous setting, if the corresponding bit in any word read from the Memory Module is a "one." Similarly, in the "test ones" mode, all "ones" may be written into any location. A new rippler register bit is set to a "one" if the corresponding bit is a "zero" when read.

It is also possible to operate the Memory in an "error-correcting mode" and in a "self-refresh mode". In the former mode, any single erroneous bit is corrected automatically whenever a word is read from the Memory Module. In the latter mode, all read accesses use the old rippler register and all write accesses use the new rippler register. In addition single bit errors are corrected as in the error-correcting mode. This permits a Memory Module with a faulty bit

line to be "repaired" by sequentially reading and then writing into each of its 4096 words. Both modes can be entered under program control by addressing the Memory Module in the hard address mode (cf. Table 5-6 for the memory hard address functions).

A memory clamp signal is triggered by signals received from the three Circumvention Modules. If at least two of these Circumvention Module signals indicate a clamped condition, the Memory Module inhibits all operation and clamps, or shunts, the word drive current so that no data in memory is lost during power failures or circumvention events.

Each Memory Module is assigned its four-bit soft name under program control (cf. Table 5-6) and responds to any properly encoded address prefixed with these four bits by accessing the location defined by the remainder of the address. In addition each module responds to the soft-mode (hexadecimal) addresses FEAX, FDDX, and FFCX (cf. Table 5-4) when "X" is its current soft name. In response to the address FEAX, the module gates the contents of its Fault Status Register onto the data bus and subsequently resets the register. Addresses FDDX and FFCX provide access to bits 32-40 and bits 0-31, respectively, of the new rippler register.

Each Memory Module also responds to all properly encoded addresses in the hard-address mode prefixed with its unique five-bit hard address. The specific response to each such address is as specified in Table 5-6. The "fault override" functions allow the memory to respond normally when the indicated faults are present. The "arm test memory monitor" functions provide a means for testing all memory monitors by simulating fault conditions. They are reset automatically upon the occurrence of the simulated fault.

5.3.4 SERIAL DATA BUS

The Serial Data Bus subsystem (Figure 5-11) provides the BFTSC with three external interfaces and allows expansion to 62 external interfaces. The basic elements of the Serial Data Bus subsystem are the dual-redundant Serial Interface Unit (SIU), which controls bus communications, and the dual-redundant Device Interface Units (DIUs) at each of the interfaces. Only one of the two SIUs is powered at a time; both of the DIUs at each interface are powered continuously.

The SIU is capable of transferring blocks of words between the serial data bus and main memory by utilizing a direct memory access capability. The SIU is also capable of transferring serial data bus words on a single word basis utilizing an interrupt capability to notify the CPU of user requests.

Transmission on the bus consists of positive and negative pulses of one-half system clock period duration separated by a one-half system clock period of inactivity. Four combinations of pulses are transmitted on the serial bus (cf. Figure 5-12): a logic 1 consisting of a positive pulse followed by a negative pulse, a logic 0 consisting of a negative pulse followed by a positive pulse, a "word sync" (preceding each word except the first word in each frame) consisting of three negative pulses followed by three positive pulses, and a "frame sync" (preceding the first word in each frame) consisting of three positive pulses followed by three negative pulses. A serial data bus frame begins with each real-time interrupt. It consists of 128 serial bus words, and has a duration of 11008 system clock cycles (for the present RTI period).

Each transmitted 40-bit serial data bus word is preceded by one of the two sync pulse sequences, neither of which is valid as normal data. The 8-bit error detection code used on the internal BFTSC data bus (cf. Section 5.1.2) is appended to the 32-bit data word transmitted on the bus. The code is maintained on each data word from Memory to SIU to DIU to SIU and back to Memory. An example of the serial data bus waveforms is illustrated in Figure 5-12.

The serial bus is a reconfigurable daisy-chain system with dual-redundant serial links between each contiguous pair of DIUs along the bus. Transmission on the bus is in one direction only. The bus data transmitted by the SIU is detected, reconstructed and repeated or modified at each DIU, thereby limiting the length of bus to be driven by any DIU transmitter.

Each of the dual DIUs provides a 20-bit parallel interface for the peripheral device with which it is associated. It receives its power from its associated peripheral device and shares the same housing. Each DIU selects one of the two serial bus lines and demodulates and decodes the received information. The DIU monitors the selected bus for faulty bus transmissions and switches to the alternate bus, if a signal is present on that bus, as soon as a fault condition is detected on the information being received. Relays are provided for bypassing a DIU pair should the associated peripheral lose power.

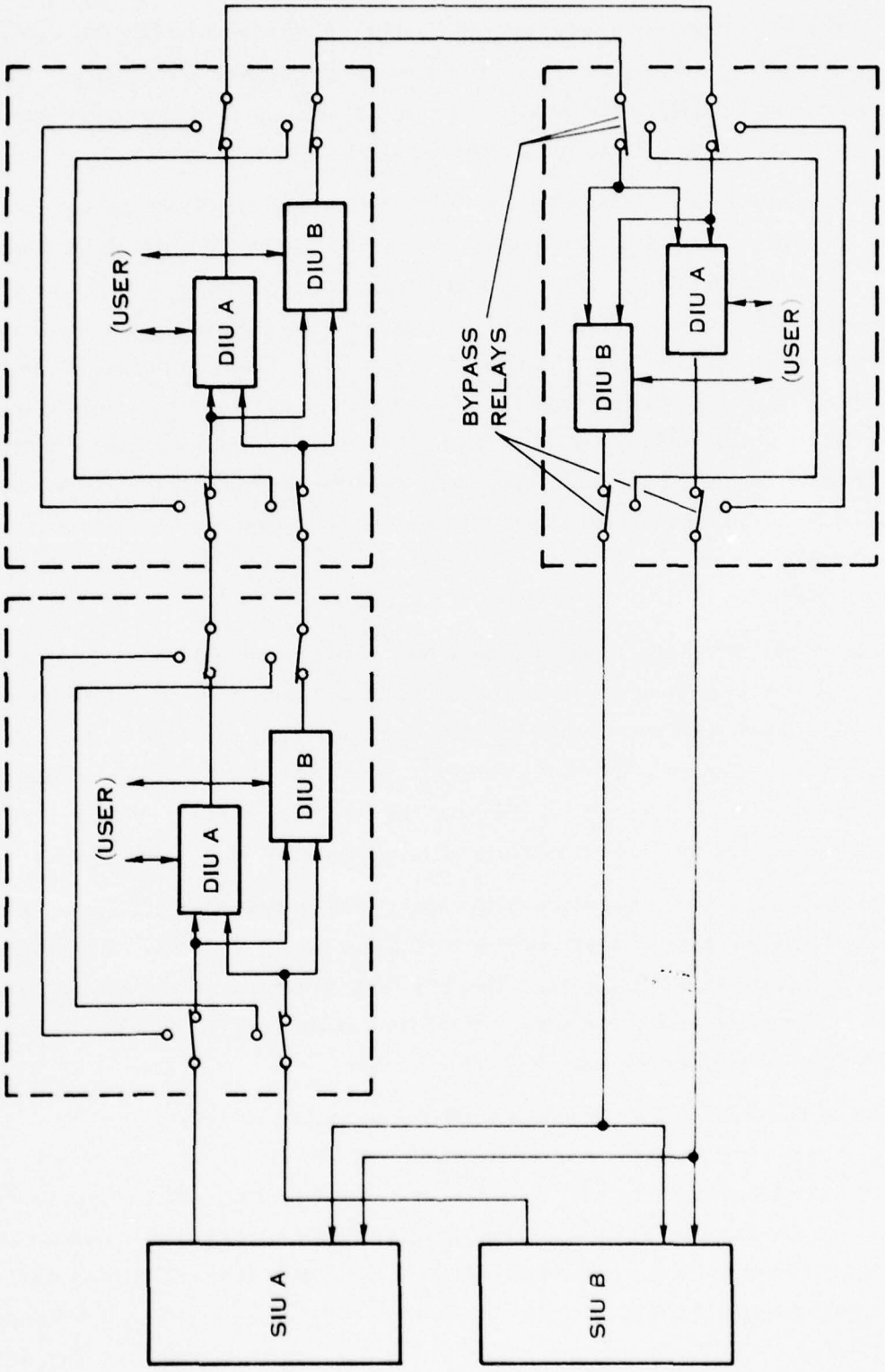


Figure 5-11. Serial Data Bus System

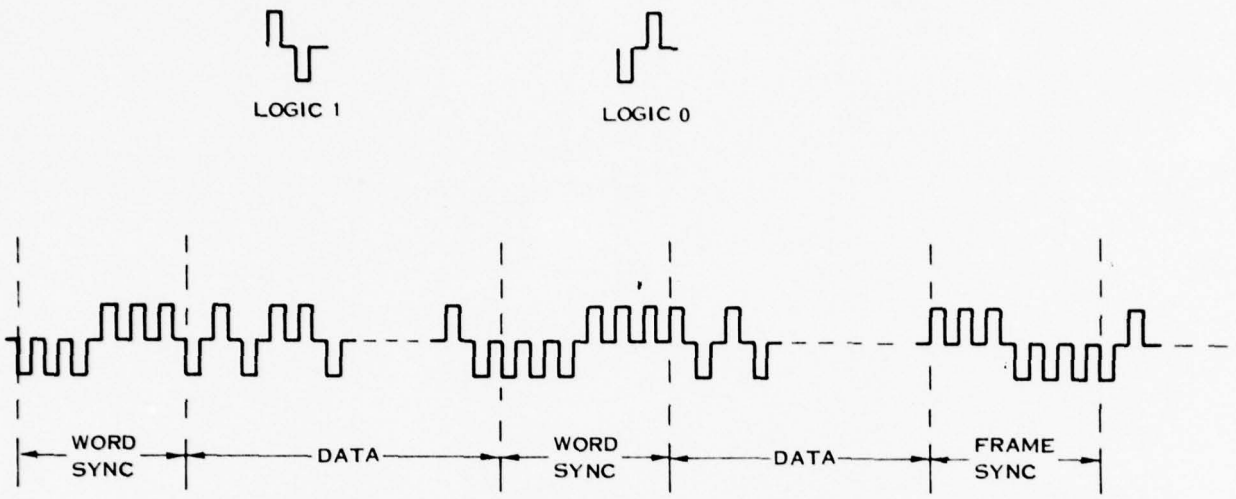


Figure 5-12. Serial Data Bus Waveforms

5.3.4.1 Serial Interface Unit (SIU)

The SIU provides the interface between the computer and the serial data bus. Figure 5-13 shows the functional features of the SIU. The SIU is program controlled with two assigned addresses. These addresses are used to send control words to the SIU (in the write mode) and to read the SIU status (in the read mode). The formats for these words are shown in Figure 5-14. The control words are used for testing purposes and to set up a block transfer, i.e., to specify the start and stop addresses in memory where the block to be transferred is located. When a data transfer is initiated, the SIU sequentially reads words from the memory locations in the specified block, places the words on the serial bus, and writes the returned, possibly altered, words back into the locations from which they were read. Normally, the block is initiated with the seventh serial data bus word after a real-time interrupt. If the "immediate" bit (cf. Figure 5-14) is set, however, the SIU begins the specified block transfer with the next available serial bus word. In either case, the "busy" bit in status word 0 is set when the block transfer begins and is reset when it has been completed.

The SIU is prevented from accessing an incorrect address by the error code incrementing scheme shown in Figure 5-15. The error code appended to the first address of the block is created by modifying the error code of the data

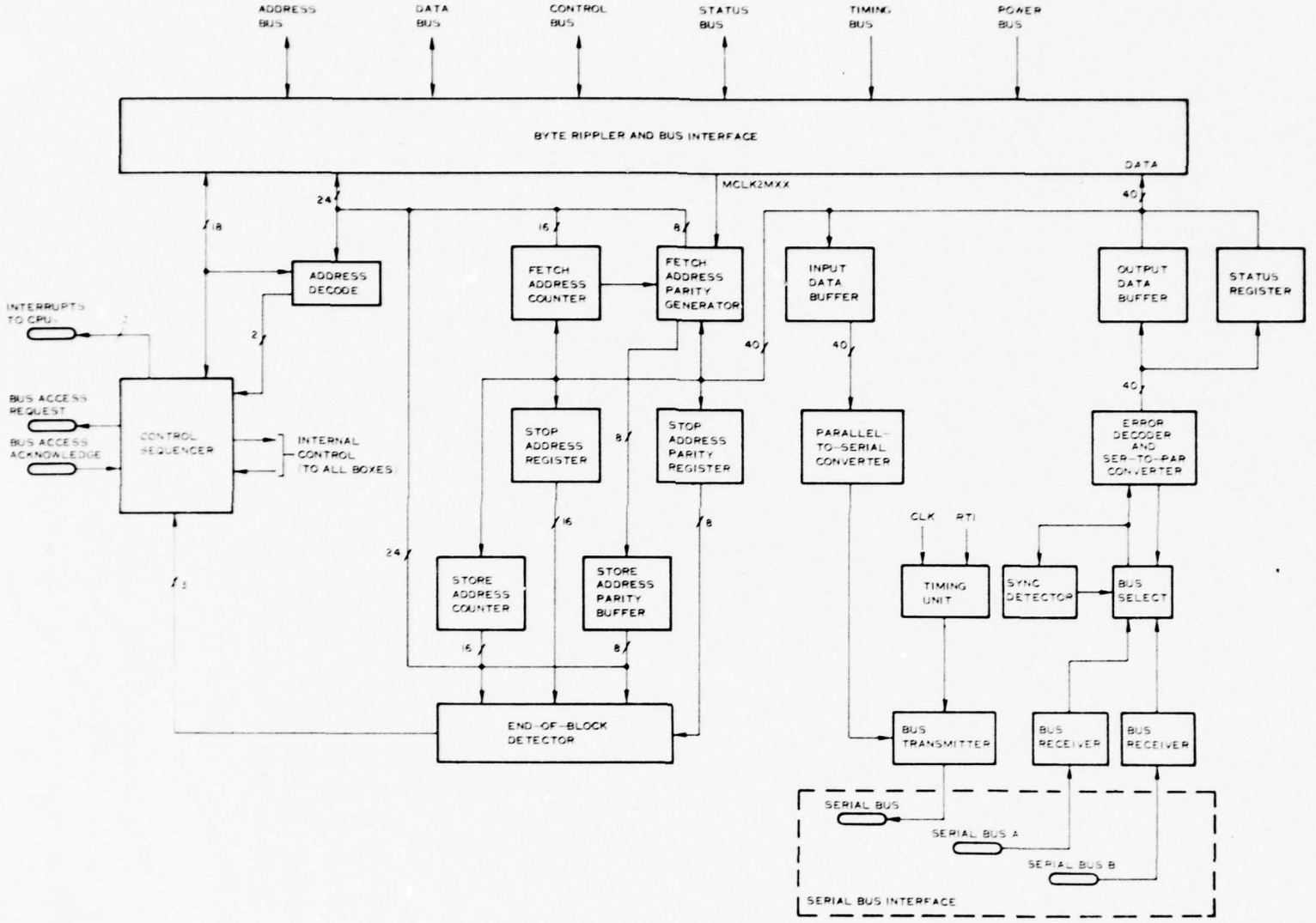


Figure 5-13. SIU Module Functional Block Diagram

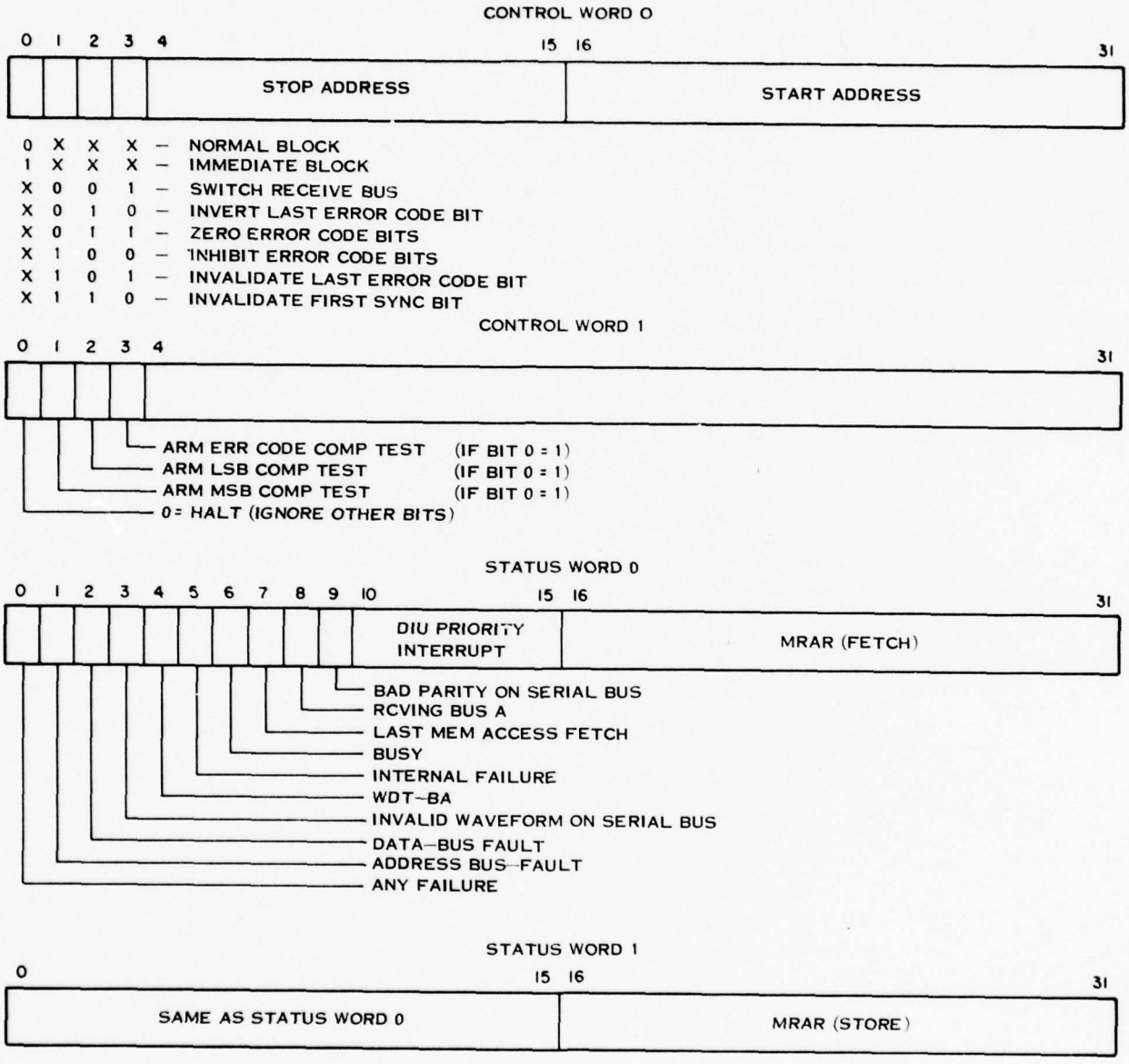


Figure 5-14. SIU Word Formats

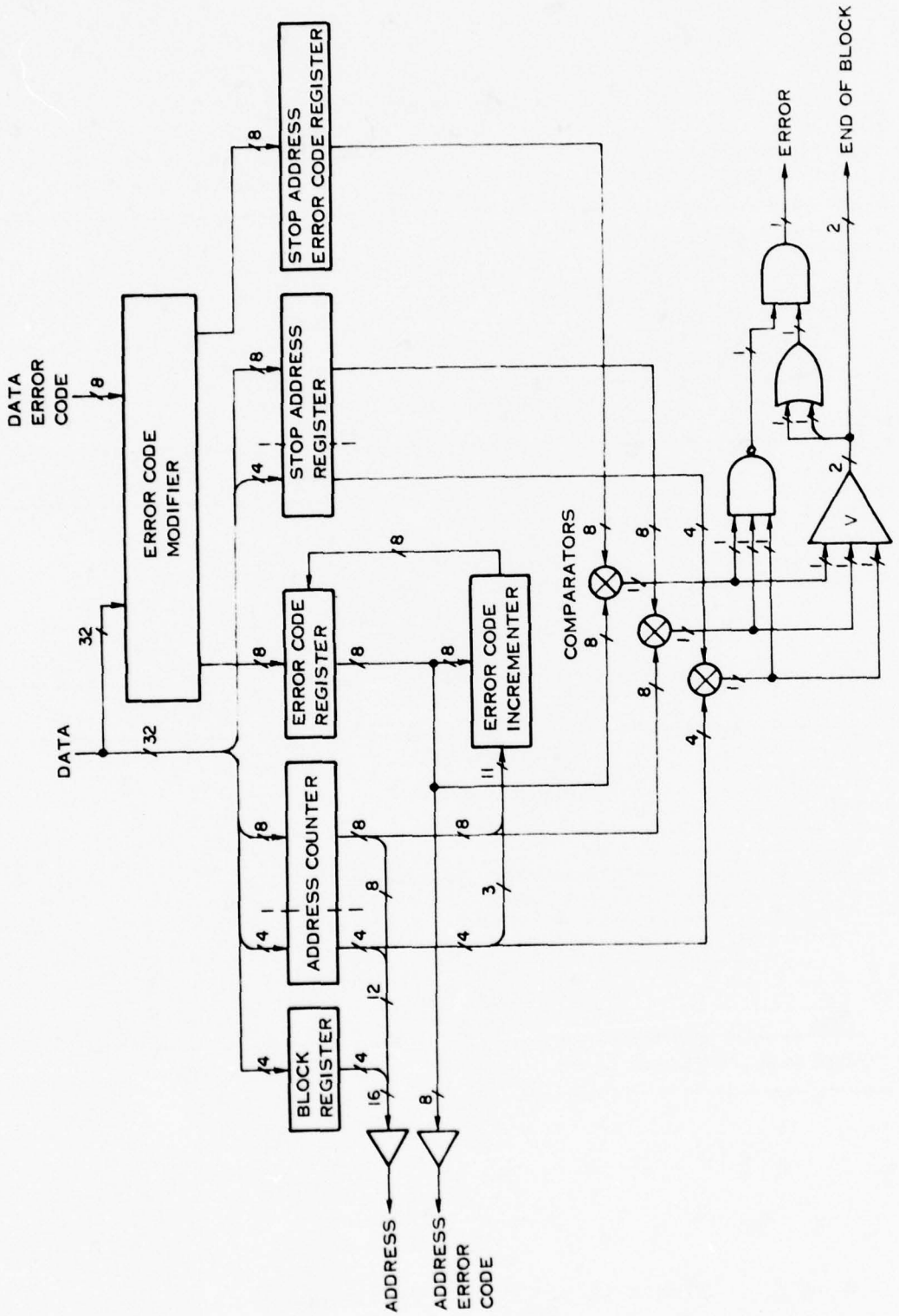


Figure 5-15. DMA/SIU Address Protection

word used to initiate the block. The error code for the next address is generated by modifying the error code of the present address using the present address. This error code scheme insures that any address accessed out of sequence by the SIU is accompanied by an incorrect error code on the address bus. Due to the capability of the error code to detect any number of failures within a single byte, end-of-block is detected when any two of the three bytes of the address and its error code in the address register match those in the stop address register. If the third byte does not match at this time, an error is indicated.

The serial data bus word format is as shown in Figure 5-16. A word is taken from main memory by the SIU, circulated on the serial data bus and stored back in the memory location from which the word was originally read. Any changes which a DIU makes to a serial data bus word being transmitted or received are made while preserving the integrity of the error-detection code. When the last word of a specified block has been stored in memory, an end-of-block (EOB) interrupt is sent to the CPU.

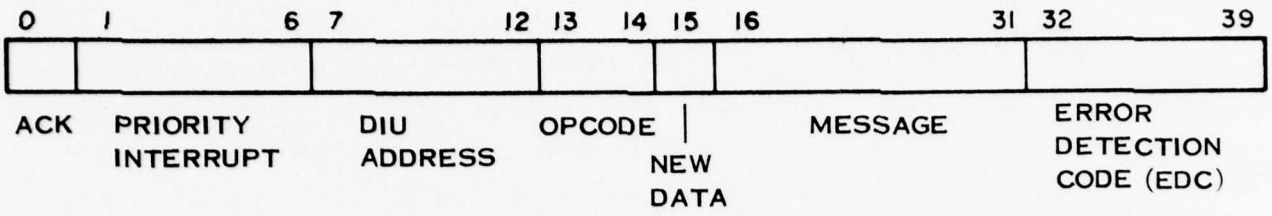
The priority-interrupt field of the serial data bus word is used for three purposes:

a. Inform the SIU that the DIU has a priority interrupt for the CPU. In this case, the ACK bit is zero and the priority interrupt field contains the address of the interrupting DIU.

b. Inform the DIU that the CPU has received the priority interrupt. In this case, the ACK bit is set to a one and the priority interrupt field contains the address of the DIU to be acknowledged.

c. Inform subsequent DIUs and the SIU that the previous word did not pass the code or waveform integrity checks (all ones in the priority interrupt field).

The SIU notifies the CPU of a DIU priority interrupt, a serial bus code or waveform integrity fault, an internal error, a lack-of-response error (from the CPU bus arbiter), or address and data bus coding errors by providing a general interrupt to the CPU. The CPU determines the cause of the interrupt by reading SIU status word 0. Also included in status word 0 is the SIU's most recent fetch address register (MRAR-Fetch) which contains the memory address most recently accessed by the SIU in the fetch mode. (The interrupt indicators and the DIU priority-interrupt field of the status word are cleared to zero when status word 0 is read by the CPU.) Status Word 1 includes the SIU's most recent store address



The ACK and Priority Interrupt fields are detailed below in order of decreasing priority.

ACK	PRIORITY INTERRUPT	
X	111111	Error detection flag. Transmitted by DIU if previous word had faulty EDC.
1 to 1	111110 to 000001	Acknowledgement. A DIU interrupt is acknowledged by transmitting a one in the ACK bit and the DIU address in the Priority Interrupt field.
0 to 0	111110 to 000001	Priority Interrupt. The DIU transmits its address in the Priority Interrupt field when computer access is desired. The DIU transmits in this field only if its priority is higher than the PI received.
X	000000	This is the normal, no interrupt, state of the PI field.

DIU ADDRESS -- Six-bit DIU address field from the SIU. Addresses 111111 and 000000 are not assigned.

OPCODE -- Instruction to DIU.

BIT 13	BIT 14	
0	0	Receive Message (SIU to DIU)
0	1	Receive Command (SIU to DIU)
1	0	Transmit Message (DIU to SIU)
1	1	Transmit Status (DIU to SIU)

NEW DATA -- The DIU sets the new data bit to a one when it accepts the word or when a transmitted message has been updated by the user since the last transmission. If the word is not accepted or the transmitted message has not been updated, the DIU sets the new data bit to a zero. A command is used only by the DIU (it is not sent to the user), if the new data bit is a one.

MESSAGE -- 16-bit message word transmitted to or from DIU depending on the indicated op code.

ERROR DETECTION CODE -- 8-bit code used for error detection on the bus. The code is identical to the byte parity code used on the data bus.

Figure 5-16. Serial Data Bus Word Format

register (MRAR-Store) containing the last memory address accessed by the SIU in the store mode.

5.3.4.2 Device Interface Unit (DIU)

5.3.4.2.1 Module Description

When addressed, a DIU (cf. Figure 5-17) either receives a message or a command, or transmits a message or a status word, as indicated by the instruction op code. (The serial data bus word format is as defined in Figure 5-16.) As a serial word is shifted into the DIU, each bit is tested to determine if it is consistent with the DIU's pin-programmed address. When the DIU detects its own address, it performs the following:

- If the op code indicates receive message and the DIU is able to accept the message, the new data bit is set to a one, the error code is modified accordingly, and the message is sent to the peripheral.
- If the op code indicates receive command, the DIU checks the new data bit. If the new data bit is a one, the DIU performs the command indicated by the first eight bits of the command word. If the new data bit is a zero and the DIU can accept the command, the new data bit is set to a one, the command is sent to the peripheral, and the command indicated by the first eight bits of the command word is performed by the DIU.
- If the op code indicates transmit message, the DIU sets the new data bit to a one and transmits the message only if the message has been updated by the user since the last time it was transmitted.
- If the op code indicates transmit status, the DIU sets the new data bit to a one and transmits the status. The first eight bits of the message field are reserved for DIU status and the second eight bits for peripheral status.

The DIU command and status word formats are defined in Figure 5-18.

5.3.4.2.2 DIU/Peripheral Interface

Data transfers between the DIU and the peripheral are implemented with twenty bi-directional data lines and nine control lines: data request, interrupt request, read/write, command/data, and half word control lines from the DIU to the peripheral, and acknowledge, device interrupt, busy, and error code control lines from the peripheral device to the DIU.

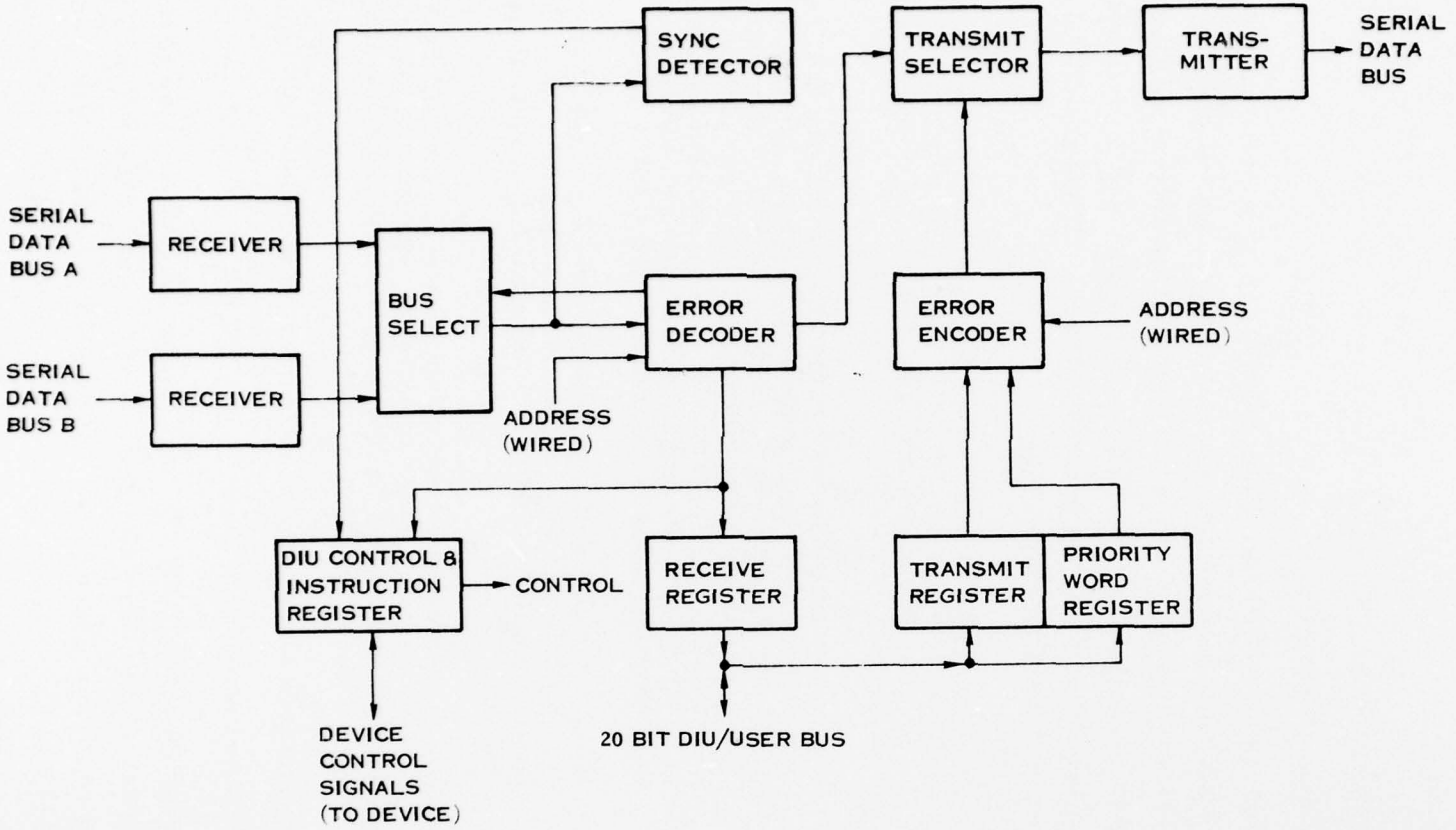


Figure 5-17. DIU Module Functional Block Diagram

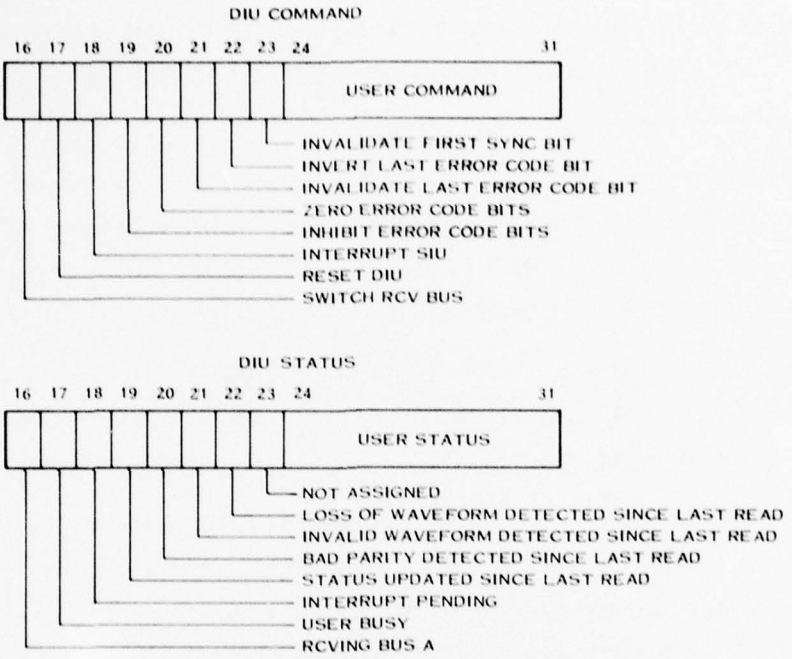


Figure 5-18. DIU Word Formats

The data request line is raised by the DIU to initiate a data transfer to or from the peripheral or to transfer a command to the peripheral. When this line is raised, the peripheral responds according to the states of the other control lines.

The read/write line indicates the direction of the transfer. It is set to the high state when the transfer is from the peripheral to the DIU and the line is set low when information is to be transmitted from the DIU to the peripheral.

The command/data line distinguishes between command information and data. If this line is at a high state, information transmitted from the DIU to the peripheral is interpreted as a command. If this line is at a low state, information transmitted in either direction is interpreted as data. Should the peripheral be using error code, the information is transferred between the DIU and the peripheral as two 20-bit half words. The half word control line is used by the DIU to indicate which half word is being transferred. If this line is low, the first half is being transferred; if it is high, the second half is being transferred. These two half words define a valid serial bus code word when combined in the following manner:

bits 0-15 of half word 1 followed by bits 0-15 of half word 2 form the 32 bits of data. Bits 16-19 of half word 1 followed by bits 16-19 of half word 2 form the eight bits of error code.

The interrupt request line is raised by the DIU in response to a device interrupt in order to transfer status from the peripheral to the DIU. (In the FTSC this line is combined with the data request line. In order to read status, the command/data line is set high and the read/write line is set high.)

The acknowledge line is raised by the peripheral after a transfer has been requested by the DIU and the peripheral has produced (gated onto the data lines) the data/status or accepted the data/command being transferred.

Peripheral initiated data transfers are accomplished using the device interrupt line. When this line is raised by the peripheral, the DIU raises its interrupt request line in order to read the status from the peripheral. The device interrupt is lowered before the transfer is completed.

The busy line is raised by the peripheral to indicate that it is unable to accept commands. When this line is high, the DIU sets the device busy bit in its status word. The status bit is reset when the busy line is lowered by the peripheral.

The error code control line from the peripheral is used to indicate whether or not error code is being transmitted. If the level on this line is high, the four least significant bits of each of the two 20-bit half words are concatenated to produce the eight code bits for the word. If this line is low, only one half word is transferred which contains the 16 message bits of the serial bus word.

5.3.5 DIRECT MEMORY ACCESS MODULE (DMA)

5.3.5.1 Module Description

The DMA channel is utilized for high-speed peripheral access to the BFTSC. The DMA (cf. Figure 5-19) is program controlled and responds to two control addresses. The first address is used to send the DMA data transfer control words and to read the first status word; the second address is used to initiate certain monitor tests, to send commands to the peripheral and to receive requests from the peripheral. (The word formats are shown in Figure 5-20.) Data transfers

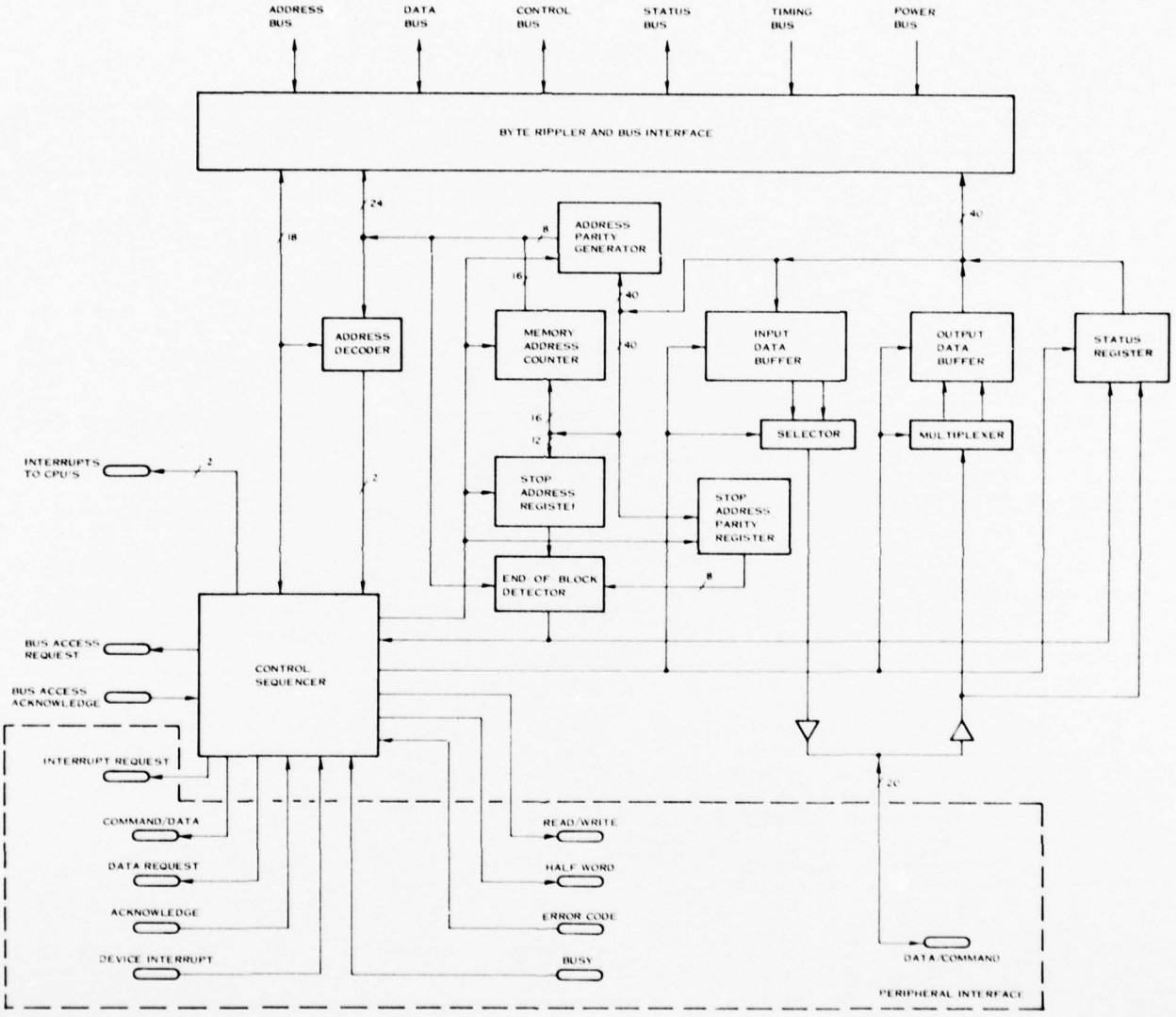


Figure 5-19. DMA Module Functional Block Diagram

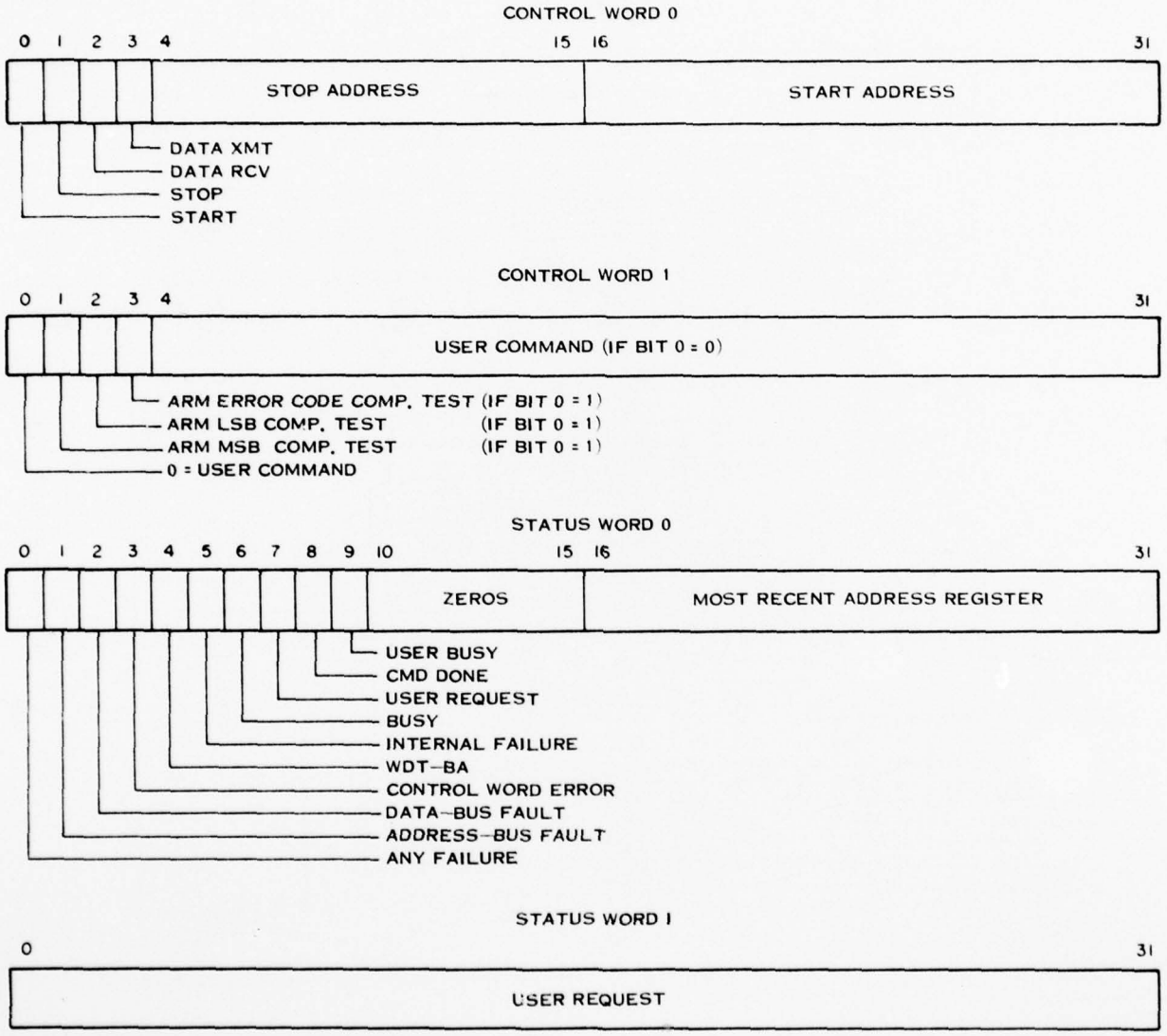


Figure 5-20. DMA Word Formats

to or from the peripheral are established by first sending commands to the peripheral (DMA control word 1).

Once the peripheral device is initialized for a data transfer, the CPU sends a command word to the DMA (DMA control word 0) to tell it the start address, the stop address, whether it is transmitting or receiving, and whether to start or stop the transfer. If the DMA is commanded to start a transfer, it sets the busy bit in its status word, and places the start address in a counter. The counter is incremented as each data word is transferred to or from memory. When the contents of this counter equal the stop address, the DMA ceases data transmission, resets the busy bit, and raises an end-of-block interrupt to the CPU.

The DMA is prevented from accessing an incorrect address by the error code incrementing scheme shown in Figure 5-15. The error code appended to the first address of the block is created by modifying the error code of the data word used to initiate the block. The error code for the next address is generated by modifying the error code of the present address using the present address. This error code scheme insures that any address accessed out of sequence by the DMA is accompanied by an incorrect error code on the address bus. Due to the capability of the error code to detect any number of failures within a single byte, end-of-block is detected when any two of the three bytes of the address and its error code in the address register match those in the stop address register. If the third byte does not match at this time, an error is indicated.

The DMA notifies the CPU of a user interrupt, a control word error, an internal error, a lack-of-response error (from the CPU bus arbiter), address and data bus coding errors, and a command transferred to the user, by providing a general interrupt to the CPU. The CPU determines the cause of the interrupt by reading DMA status word 0. Also included in status word 0 is the most recent address register (MRAR) which contains the memory address most recently accessed by the DMA. (The interrupt indicators in the status word are cleared to zero when it is read by the CPU.) In the case of a user interrupt, status word 1 contains the user status.

5.3.5.2 DMA/Peripheral Interface

Data transfers between the DMA and the peripheral are implemented with twenty bi-directional data lines and nine control lines: data request, interrupt request, read/write, command/data, and half word control lines from the DMA to the peripheral, and acknowledge, device interrupt, busy, and error code control lines from the peripheral device to the DMA.

The data request line is raised by the DMA to initiate a data transfer to or from the peripheral or to transfer a command to the peripheral. When this line is raised the peripheral responds according to the states of the other control lines.

The read/write line indicates the direction of the information transfer. It is set to the high state when the transfer is from the peripheral to the DMA; the line is set to the low state when the transfer is from the DMA to the peripheral.

The command/data line distinguishes between command information and data. If this line is at a high state, then information transmitted from the DMA to the peripheral is interpreted as a command. If this line is at a low state, information transmitted in either direction is interpreted as data.

Information transferred between the DMA and the peripheral is transmitted as two 20-bit half words. The first half word consists of the 16 most significant data bits and the four most significant error code bits of the 40-bit internal data bus word. The second half word consists of the 16 least significant data bits and the four least significant error code bits of the word. The half word control line indicates, when low, that the first half word is being transferred and, when high, that the second half word is being transferred.

The interrupt request line is raised by the DMA in response to a device interrupt in order to transfer status from the peripheral to the DMA. (In the FTSC this line is combined with the data request line. In order to read status, the command/data line is set high and the read/write line is set high.)

The acknowledge line is raised by the peripheral after a transfer has been requested by the DMA and the peripheral has produced (gated onto the data lines) the data/status or accepted the data/command to be transferred.

Peripheral initiated data transfers are accomplished using the device interrupt line. When this line is raised by the peripheral, the DMA raises its interrupt request line in order to read the status from the peripheral. The device interrupt is lowered before the transfer is completed.

The busy line is raised by the peripheral to indicate that it is unable to accept commands. When this line is high, the DMA sets the device busy bit in its status word. The status bit is reset when the busy line is lowered by the peripheral.

The error code control line is used by the peripheral to indicate whether or not information transmitted from the peripheral to the DMA has been encoded. If the level on this line is high, the four least significant bits of each of the two 20-bit half words are concatenated to produce the eight code bits transmitted on the internal data bus. If this line is low, the DMA ignores the four least significant bits in each of the two 20-bit half words and internally generates and appends the appropriate code bits.

5.3.6 TIMING MODULE

The BFTSC contains two Timing Modules, each of which includes an oscillator, countdown chain and buffers, dual redundant monitors, and interface logic. The configuration of the Timing subsystem is as shown in Figure 5-21. The output of each Timing Module drives a separate Timing bus.

Both Timing Modules are powered in order to minimize reconfiguration time delays. However, only one Timing Module is in the active mode at any time. The second module serves as a standby spare.

The active Timing Module generates three coherent timing signals:

a. An external real-time clock (EXRTC).

b. A real-time interrupt (RTI) every 8.8064 milliseconds. It is possible to increase the RTI period by a factor of two or decrease it by a factor of two or four by selecting different RTI taps.)

c. A system timing clock (CLK) at a frequency of 1.25 MHz.

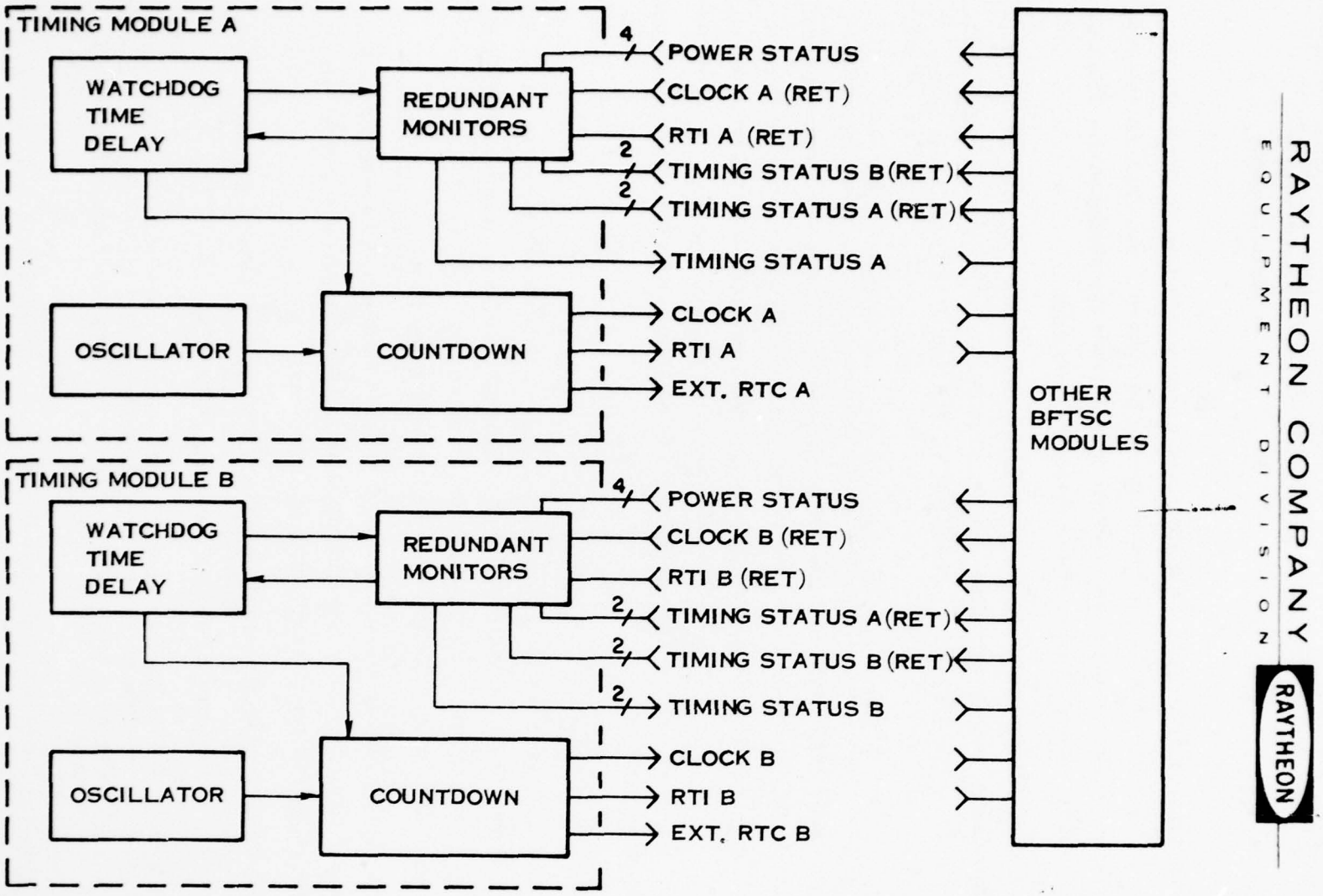


Figure 5-21. Timing Subsystem Functional Block Diagram

Dual-redundant monitors are provided for each timing output which examine the outputs at the end of the Timing bus, i. e., where the bus is returned to the Timing Module. Each of these monitors uses a status line to announce the status of its associated module to the rest of the BFTSC system. If either of the status lines associated with the active module should indicate an out-of-tolerance condition while the spare module is functioning properly, the latter raises its status lines causing all BFTSC modules to switch over to it, thus reversing the roles of the active and spare modules. The Timing Modules are biased such that when the power status lines become valid, Timing Module A raises its status lines before Timing Module B. When one Timing Module lowers its status lines as the result of a failure, that module is prevented, by a watchdog timer, from raising its status lines again for at least 12 clock pulses. This is sufficient time for the alternate Timing module to raise its status lines.

5.3.7 POWER MODULE

The BFTSC Power Distribution System (cf. Figure 5-22) consists of two redundant Power Modules, each of which is comprised of an EMI/RFI-EMP protection network, switch assembly, start-up and shut-down control circuitry, and redundant monitors for all output voltages. Both modules use the regulated outputs from a single set of commercial supplies and energy storage unit. (In the FTSC, each Power Module shall have its own regulators and energy storage unit and shall be powered by a single input voltage of 22-30 Vdc.)

Each output-voltage monitor is dual-redundant and examines the voltages at the end of the power distribution system bus. Each of these monitors uses a status line to announce the status of its associated module to the rest of the BFTSC system. If either of the status lines associated with the active module should indicate an out-of-tolerance condition, a switchover to the alternate module is effected.

The energy storage unit provides sufficient energy to power critical circuits such as the Hardened Timer, Circumvention System, Timing Module oscillators, and all monitor circuits for a period of 60 milliseconds after the power conditioner outputs have gone out of tolerance.

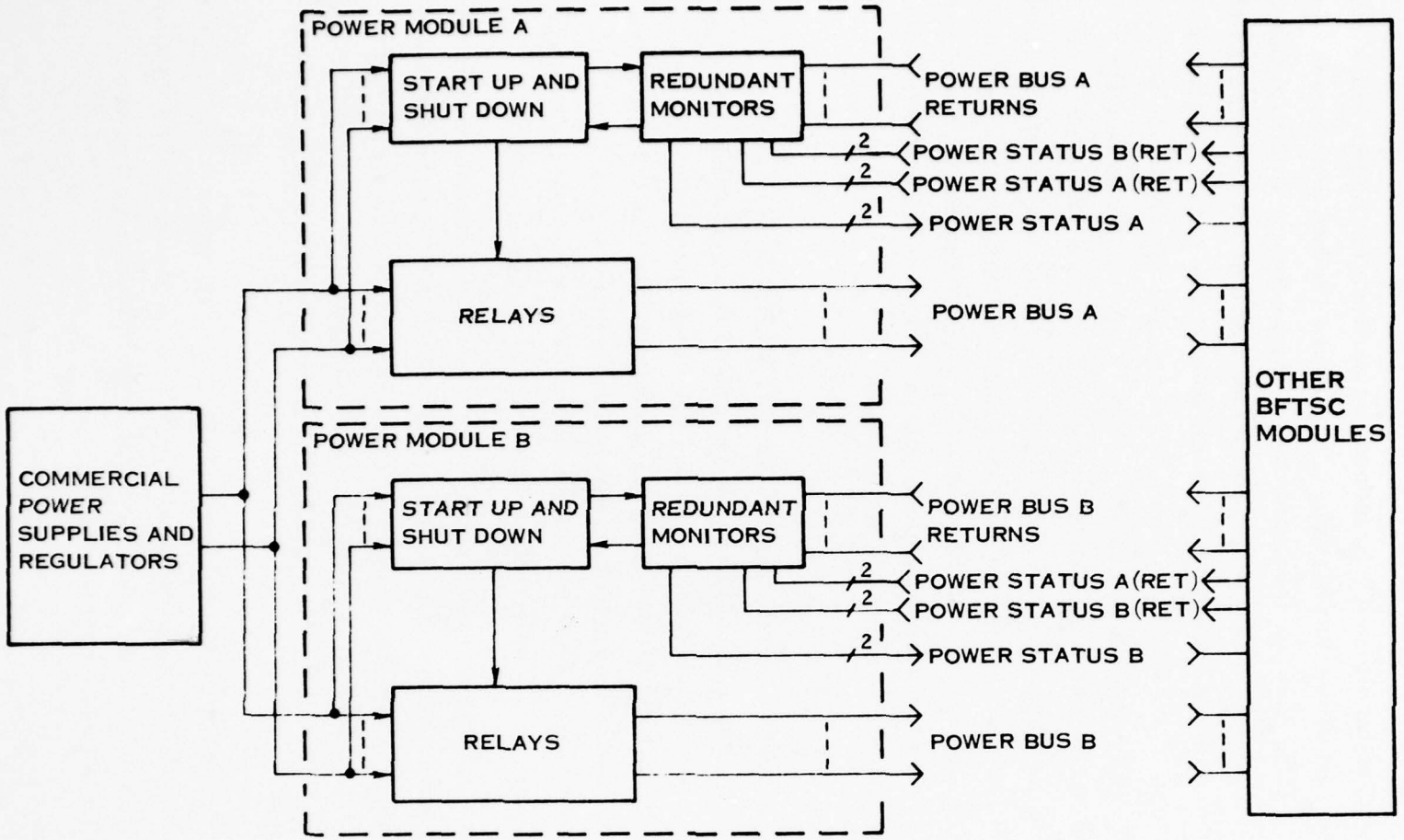


Figure 5-22. Power Subsystem Functional Block Diagram

5.3.8 CIRCUMVENTION MODULE

The BFTSC contains three Circumvention Modules acting in a triple-modular redundant configuration. Each Circumvention Module (cf. Figure 5-23) has a provision for accepting inputs from a set of remote radiation detectors.

The Circumvention Modules each provide a "zero" level on the memory clamp and I/O clamp lines under any of the following conditions:

a. Neither Power Module is operating within tolerance;

b. Neither Timing Module is operating within tolerance;

c. The radiation level exceeds the threshold of any of the radiation detectors associated with that Circumvention Module.

In addition, each Circumvention Module provides a "zero" level circumvention signal to each CCU if either condition 1. or condition 3. is satisfied.

The memory clamp, I/O clamp and the CCU circumvention level signals are maintained for 10 milliseconds after the radiation falls below the threshold of the detectors or the power status lines return to the "one" level.

5.3.9 HARDENED TIMER

The BFTSC contains three Hardened Timers operating in a triple-modular-redundant configuration. Each Hardened Timer keeps track of real time during circumvention, power-down and reconfiguration periods. The Hardened Timer configuration is as shown in Figure 5-24.

A "zero" level on any two of the I/O clamp signal lines from the three Circumvention Modules activates each Hardened Timer. Each Hardened Timer is also activated by reconfiguration mode signals from any two of the three CCUs.

A downcount hardened timer signal from the CPU, in the absence of any activation signals, causes the Hardened Timer to cease charging the capacitor and to begin discharging at a rate 100 times faster than the charge rate. The RTI output flop is set once each 1/101st RTI until the capacitor is completely discharged or until it is reactivated. The Hardened Timer is able to record an elapsed time of up to at least six seconds.

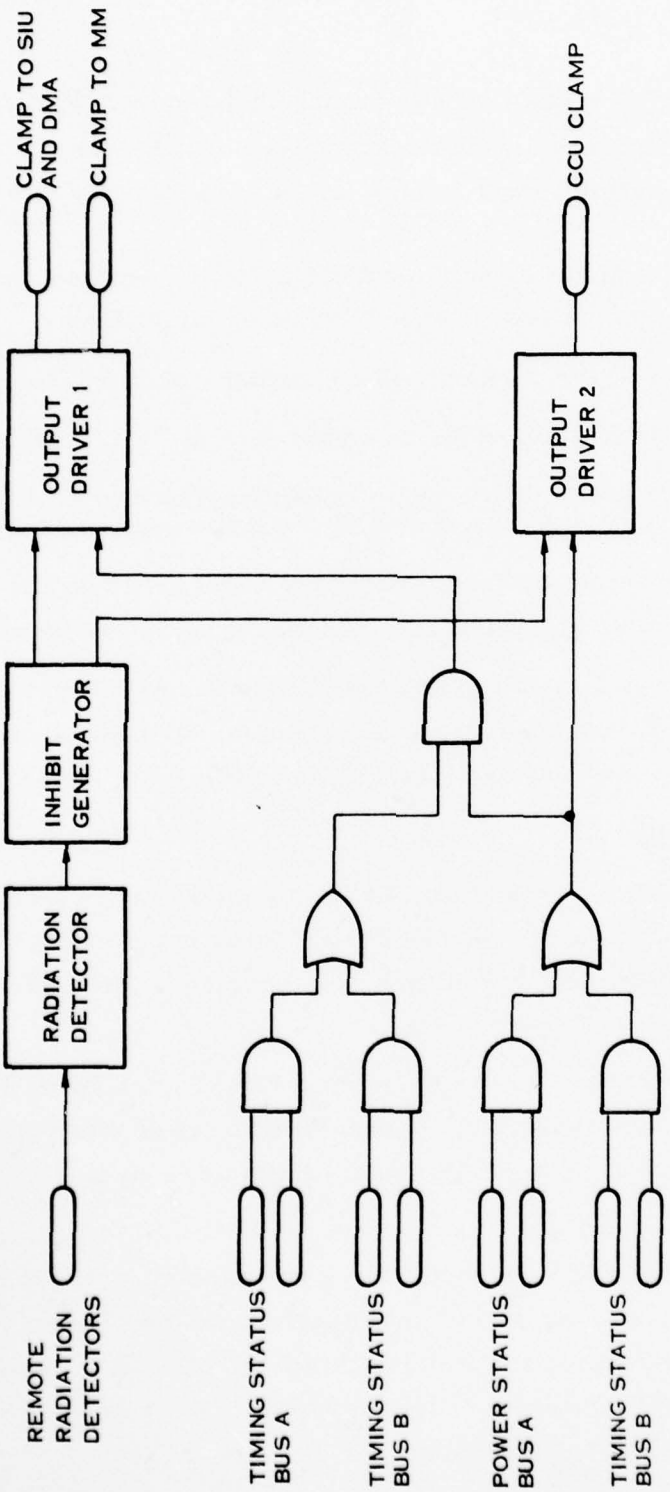


Figure 5-23. Circumvention Module Functional Block Diagram

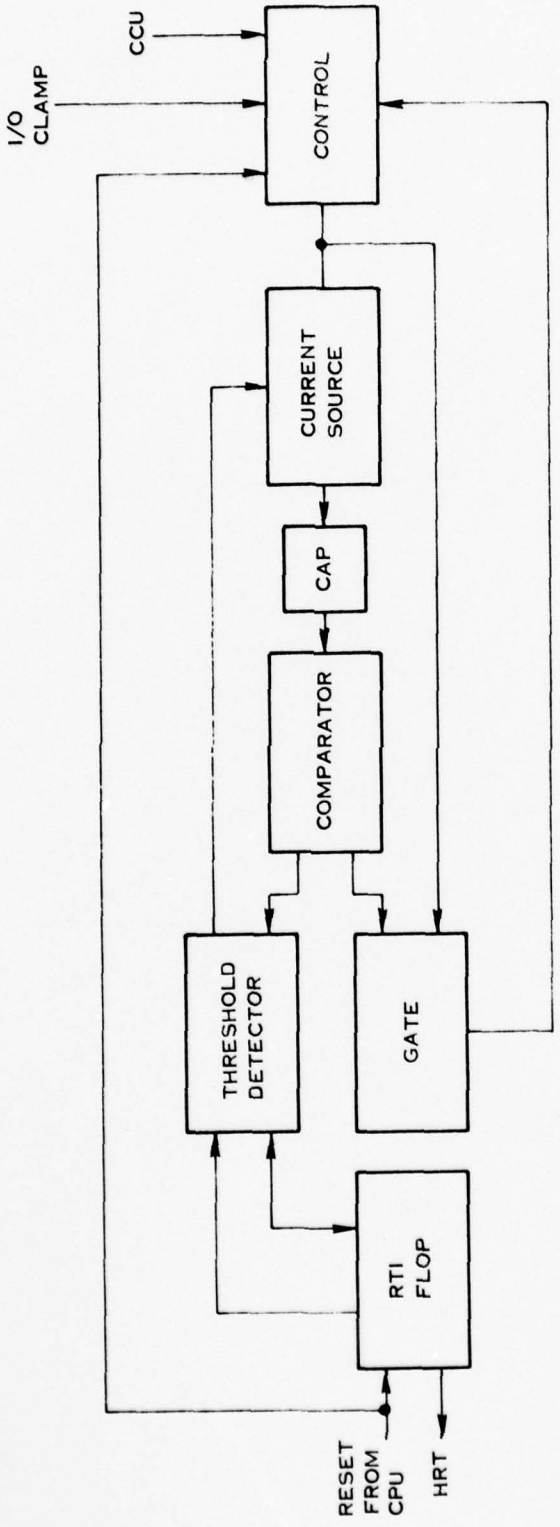


Figure 5-24. Hardened Timer Module Functional Block Diagram

SECTION 6

FAULT DETECTION MECHANISMS

The BFTSC contains a distributed set of hardware monitors for the purpose of fault detection and isolation. The hardware monitors operate concurrently with normal system operation and are transparent to the system user. All hardware monitors are either capable of being tested under program control or provided redundantly to ensure their operation.

6.1 ADDRESS AND DATA BUS ERROR CODE MONITORS

Address and data are each accompanied by an eight bit error code (cf. Paragraph 5.1.1 and 5.1.2) when transmitted by a module on the internal address or data bus. Each CPU, DMA, SIU, and Memory Module has an address and data decoder which computes the error code for each word transmitted and compares this computed code to the code received with the word. Any address or data code discrepancy detected by a CPU, DMA, or SIU Module is reported (along with its type) directly to the CCU. A discrepancy detected by a Memory Module causes that module to inhibit its response to the request, resulting in a CPU watchdog timer alarm (cf. Section 6.2).

6.2 CPU WATCHDOG TIMER

Each CPU contains a watchdog timer which measures the time interval between the instant a device (CPU, DMA, SIU) is given access to the bus system by the bus arbiter and the instant this access is relinquished by the device (cf. Section 5.1.3). If access has not been relinquished within 10 clock periods, a watchdog timer alarm fault is generated and reported directly to the CCU.

6.3 CPU MONITOR

The monitor CPU performs all calculations and generates all outputs in synchronism with the active CPU but does not transmit its outputs on the internal buses (with the exception of the program flags and the monitor fault signal which are transmitted to the CCU). The monitor CPU compares all of its outputs (except program flags and monitor fault) to those received over the buses from

the active CPU and reports any discrepancy directly (via the monitor fault signal) to the CCU. The monitor CPU also contains a monitor mask register which is used, under program control (cf. Section 5.3.1), to invert any output from the monitor CPU for the purpose of testing the monitor comparators.

6.4 CPU FLAG MONITOR

Each CCU receives three program flag inputs from each CPU. The inputs from the monitor CPU are compared to those received from the active CPU and any discrepancy is reported to the fault decode logic within the CCU. The CCU also monitors the program flag inputs for any illegal sequence of flags (cf. Section 5.3.2) and reports this condition to the fault decode logic within the CCU.

6.5 ILLEGAL OP CODE MONITOR

Each CPU monitors the op code in the instruction register and compares it to a prewired maximum assigned op code. If the op code is greater than the maximum, an illegal op code interrupt is generated within the CPU.

6.6 STOP ADDRESS MONITOR

Each DMA and SIU Module contains a monitor to ensure that the end-of-block detectors are operating correctly. The stop address is stored in the DMA and SIU along with its error code. Three comparators (one for each byte) compare bits 0 to 7 of the address counter, bits 8 to 15 of the address counter, and the error code associated with the address counter to the corresponding bits in the stop address register. Since the error code is capable of detecting any number of failures within a single byte, any two comparisons are sufficient to detect end-of-block although all three should compare at this time. If the third byte does not compare, a failure is reported to the CPU via the DMA or SIU general interrupt and a bit in the DMA or SIU status register.

6.7 BUS ARBITER WATCHDOG TIMER

Each DMA and SIU Module contains a watchdog timer which measures the time interval between the instant it requests access to the bus system and the instant that access is granted. If access has not been granted within 31 clock periods, a failure is reported to the CPU via the DMA or SIU general interrupt and a bit in the DMA or SIU status register.

6.8 CONTROL WORD MONITOR

Each DMA and SIU module contains a control word monitor to prevent a block from being initiated or a command from being sent while a block or command is already being processed. In the DMA, the monitor also ensures that the start and stop bits in control word 0 (cf. Figure 5-20) are not both zeros or both ones. If the control word monitor detects a failure, it is reported to the CPU via the DMA or SIU general interrupt and a bit in the DMA or SIU status register.

6.9 SERIAL BUS WAVEFORM MONITOR

Each SIU and DIU Model continuously monitors the serial bus. If an invalid waveform is detected by a waveform monitor, the SIU or DIU switches to receive on the alternate bus. The SIU reports such a failure to the CPU via the SIU general interrupt and a bit in the SIU status word. Any SIU or DIU can be commanded, under program control (cf. Figures 5-14 and 5-18), to transmit an invalid waveform in order to test the monitors in the next module on the bus.

6.10 SERIAL BUS ERROR CODE MONITOR

Each word transmitted on the serial bus is followed by an 8-bit error code (the same code as that used on the internal data bus). Each SIU and DIU computes the error code for each word as it is shifted in, and compares this code with the code received with the word. Any discrepancy detected by a monitor, causes that module to switch to the alternate bus. The failure is reported to the CPU via the SIU general interrupt and bits in the SIU status word. Any SIU or DIU can be commanded, under program control (cf. Figures 5-14 and 5-18), to transmit an incorrect error code in order to test the monitors in the following modules on the bus.

6.11 SOFT NAME MONITOR

Each Memory Module has two independently loadable soft name registers, each of which is compared to the received address, to decode the module's soft address. If the outputs of the two decoders do not agree, the soft name monitor inhibits the Memory Module from responding to the request and sets a bit in the memory status word. This lack of response causes a watchdog timer fault report from the CPU to the CCU (cf. Section 6.2). The soft name monitor can be tested by loading the two soft name registers with different soft names (cf. Table 5-6) under program control.

6.12 WRITE PROTECT MONITOR

The write protect monitor in each Memory Module checks the address with the write protect register. If an attempt is made to write into a write protected quadrant of the module, the monitor sets a bit in the memory status word and inhibits response to the request. This causes a CPU watchdog timer fault report to the CCU. The write protect register can be loaded under program control (cf. Table 5-6) in order to test the monitor.

6.13 ADDRESS DECODE MONITOR

In order to ensure that the correct word select line has been enabled once the address is decoded, the address decode monitor in the memory senses the word select line and computes the error code (6-bit memory code) associated with the address of that line. This code is compared with the error code (converted from 8-bits to 6-bits) of the address received from the bus. If a discrepancy is detected, a bit is set in the memory status word and memory response is inhibited causing a CPU watchdog timer fault report to the CCU. In order to test this monitor, the address error code received from the bus (8-bits) can be forced to all zeros under program control (cf. Table 5-6).

6.14 MEMORY DATA ERROR CODE MONITOR

The memory data error code monitor performs two comparisons on the data read from or written to the memory. The error code from the system bus encoder/decoder (after conversion from 8-bits to 6-bits) and the error code stored with the data in memory are each compared to an independent 6-bit code computed from the data. (On a write, the two comparisons are identical.) Any discrepancy sets the appropriate bit(s) in the memory status word and either inhibits response to or extends the acknowledgment of the request (depending upon when the discrepancy is detected) thereby causing a CPU watchdog timer fault report to the CCU. The memory can be set up, under program control, to accept (and store) improperly encoded data for the purpose of testing the error code monitors.

6.15 SYNDROME MONITOR

When a Memory Module is in the error correcting mode (cf. Section 5.3.3), the first discrepancy detected between the computed error code and the code stored with the data causes the syndrome ("EXCLUSIVE OR" of the two codes) to be saved in the syndrome register. If any subsequent error code discrepancy produces a different syndrome, the syndrome monitor sets a bit in the memory status word and inhibits response to the request which causes a CPU watchdog timer fault report to the CCU.

6.16 REFRESH REQUEST MONITOR

When a Memory Module is in the self-refresh mode (cf. Section 5.3.3), the refresh request monitor verifies that the Memory Module is alternately read from and then written to. If two successive reads or writes occur, a bit is set in the memory status word and memory response is inhibited causing a CPU watchdog timer fault report to the CCU.

6.17 RIPPLER OVERFLOW MONITOR

Each Memory Module has a rippler overflow monitor to detect when an attempt is made to ripple out more than three bits. If an overflow is detected, a bit is set in the memory status word and either response is inhibited or acknowledgement is extended causing a CPU watchdog timer fault report to the CCU.

6.18 CHANNEL SELECT MONITOR

Each Plated Wire Memory Module is constructed using four stacks or channels. Selection of the appropriate channel is accomplished through the use of two decoders whose outputs are compared at the end of a loop connecting them to their destinations. If a discrepancy in the two decoder outputs is detected, a bit is set in the memory status word and acknowledgement is extended causing a CPU watchdog timer fault report to the CCU. One of the decoders can be disabled under program control for the purpose of testing the monitor.

6.19 X/Y SWITCH MONITOR

Each Plated Wire Memory Module uses a matrix technique to select the word to be accessed. A word is selected by enabling one X switch and one Y switch. The X/Y switch monitor checks the switches before the address decode

is enabled to ensure that all switches are off and after the address decode is enabled to ensure that both an X switch and a Y switch are on. If an invalid condition is detected, a bit is set in the memory status word and acknowledgement is extended causing a CPU watchdog timer fault report to the CCU. The monitor can be tested by invalidating the switch sense lines under program control.

6.20 WORD CURRENT MONITOR

Each Plated Wire Memory Module contains a word current monitor which senses the current through the X/Y matrix. If no current is detected when the word current is enabled, a bit is set in the memory status word and acknowledgement is extended causing a CPU watchdog timer fault report to the CCU. The word current can be inhibited under program control in order to test the monitor.

6.21 MEMORY TIMING PULSE MONITOR

Each Plated Wire Memory Module contains a timing pulse monitor which verifies that all of the required timing signals occur for each memory cycle. If a signal is missing, a bit is set in the memory status word and acknowledgement is extended causing a CPU watchdog timer fault report to the CCU. Each of the timing pulses can be inhibited under program control in order to test the monitor.

6.22 TIMING BUS MONITOR

The system clock, real-time interrupt, and timing status lines from each Timing Module are routed in a loop from the Timing Module to each module which uses them and then back to the Timing Module. There are dual-redundant timing bus monitors in each Timing Module, each of which controls one timing status line. If a monitor detects an out-of-tolerance condition on any of the received lines, it lowers the timing status line associated with it thus reporting the fault directly to the CCU.

6.23 POWER BUS MONITOR

The BFTSC power lines and power status lines from each Power Module are routed in a loop from the Power Module to each module which uses them and then back to the Power Module. There are dual-redundant power bus monitors in each Power Module, each of which controls one power status line. If a monitor

detects an out-of-tolerance condition on any of the received lines, it lowers the power status line associated with it. The Circumvention Unit monitors the power status lines and reports this condition directly to the CCU.

6.24 RADIATION DETECTORS

Each Circumvention Module accepts inputs from a set of radiation detectors (simulated in the BFTSC) distributed throughout the system. When a radiation event is detected, the Circumvention Module inhibits all I/O and memory operations and reports the fault directly to the CCU.

APPENDIX A

INSTRUCTION SET AND DATA FORMATS

A.1 DATA FORMATS

The BFTSC data formats are shown in Figure A-1 and consist of logical fixed-point (integer), and floating-point types. There is no checking of data types during execution. The instruction operand specifies the data type to be used. Hence, if a floating-point add instruction is executed, the data will be manipulated as floating-point type without checking the type.

A.1.1 Immediate Data Format

Immediate data words are used as operands for instructions in the immediate mode (cf. Section A.3). These data words occupy the 16 LSBs (bits 16-31) of the instructions. Bit 16 (the sign bit) of immediate words is extended through to bit 0 to produce the 32-bit operands used in the computations.

A.1.2 Logical Data Format

Logical data words consist of 32 bits (0-31), which are treated as unsigned words by the logical instructions.

A.1.3 Fixed-Point Format

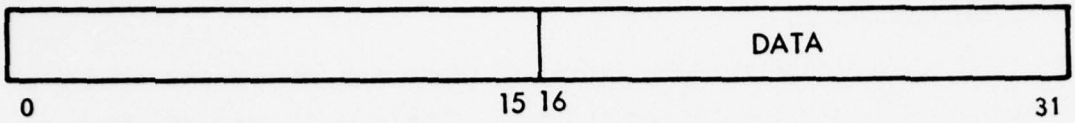
Fixed-point numbers are represented as 32-bit, two's-complement integers. The binary point is located to the right of the LSB (bit 31). The range of integers representable is:

$$-2^{31} \leq N \leq 2^{31}-1$$

A.1.4 Floating-Point Format

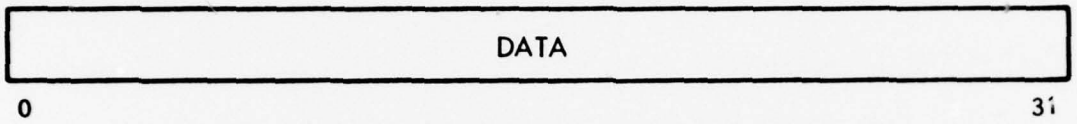
Floating-point numbers are represented by 32-bit words consisting of a 24-bit, two's complement, fractional mantissa and an 8-bit, two's complement, integer exponent. The mantissa occupies the upper 24-bits of the word (bits 0-23) and the exponent occupies the lower 8 bits (24-31). The binary point for the fractional mantissa is to the right of the sign bit (bit 0). A normalized floating

IMMEDIATE DATA



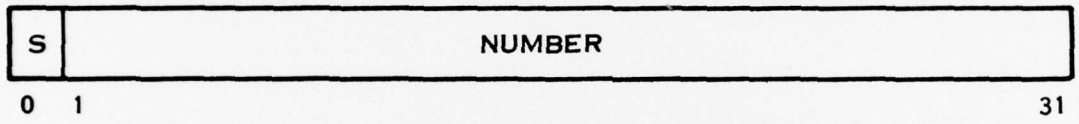
Bit 16 is extended by the immediate instructions to the upper 16 bits (0 - 15) of the word to form a 32-bit operand.

LOGICAL DATA



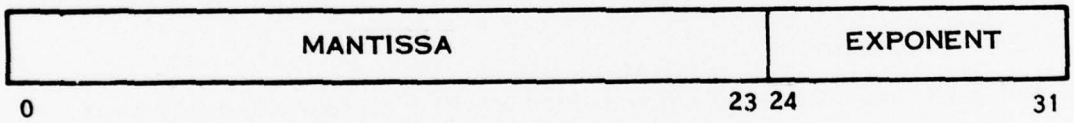
All bits are treated identically.

FIXED POINT NUMBER



Fixed-point numbers (integers) are represented in a 32-bit, two's-complement form.

FLOATING POINT NUMBER



Floating-point numbers have a 24-bit, two'scomplement, fractional mantissa and an 8-bit, two's-complement exponent. Zero is represented as a word of all-zeros mantissa and an exponent of 80 hex.

Figure A-1. Data Formats

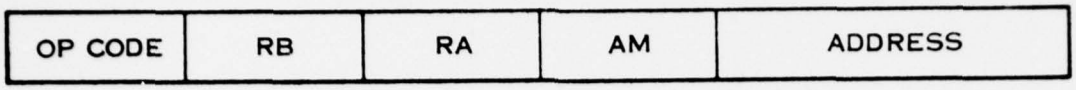
point number has the sign bit of the mantissa (bit 0) different from the next most significant bit (bit 1). The numeric value zero is represented as an all-zero mantissa and a maximum negative exponent (80_{hex}). All floating-point instructions assume normalized inputs and produce normalized results.

A.2 INSTRUCTION FORMAT

The BFTSC utilizes a single instruction format with capability to define up to 128 operation codes (op codes). Op codes 00 hex to 3F hex are classed as fetch-type instructions and the remaining op codes (40 hex to 7F hex) are classed as store-type instructions. The instruction format is shown in Figure A-2.

The following legend is used throughout this appendix:

op code	Operation Code
RB	Instruction RB Field
RA	Instruction RA Field
AM	Addressing Mode Field
ADDRESS	Instruction Address Field
r_b	Register specified in RB Field
r_{b+n}	r_b + n (modulo 8)
r_a	Register specified in RA Field
am	Addressing Mode specified in AM Field
a	Contents of ADDRESS Field
ea	Effective address
(x)	Contents of x
x, y	x concatenated with y
x ← y	x is replaced by y
ex	Extension register
Wn	Working register n
PC	Program Counter
OP	Operand



Field	Bits	Meaning
OP Code	0 - 6	Operation code
RB	7 - 9	Specifies one of 8 source/destination registers
RA	10 - 12	Specifies one of 8 source/destination/ address/index registers
AM	13 - 15	Specifies addressing mode
Address	16 - 31	Specifies a source or destination address

Figure A-2. Instruction Format

A.3 ADDRESSING MODES

The "effective address" for an instruction is determined by combining the contents of the ADDRESS field and the register specified by the RA field according to the specified address mode as defined in Table A-1. For fetch-type instructions, the effective address specifies the source of one of the operands for the instruction. The other operand (for dyadic instructions) is contained in r_b. The destination for the result is r_b unless otherwise specified by the instruction. For store-type instructions, the effective address specifies the destination for the result or the location of the next instruction to be executed. The source location is r_b unless otherwise specified by the instruction.

A.3.1 Direct Modes

Mode 0 - Register to register. The source of the operand for fetch-type instructions is r_a. For store-type instructions, the result is placed in r_a (except for jump instructions for which r_a contains the jump address).

TABLE A-1

ADDRESS MODES

Mode	Meaning	Instruction Type	Source/Destination Rules
0	Register - Register	Fetch	OP = (ea) = (r_a)
1	Immediate	Fetch	OP = (ea) = a
2	Direct	Fetch	OP = (ea) = (a)
3	Indirect	Fetch	OP = (ea) = ((a))
4	Indexed, Postincrement	Fetch	OP = (ea) = (a + (r_a)) $r_a \leftarrow r_a + 1$
5	Indexed, Predecrement	Fetch	$r_a \leftarrow r_a - 1$ OP = (ea) = (a + (r_a))
6	Indexed	Fetch	OP = (ea) = (a + (r_a))
7	Indexed, Indirect	Fetch	OP = (ea) = ((a + (r_a)))
0	Register - Register	Store	ea = r_a
1	Immediate	Store	ea = a
2	Direct	Store	ea = a
3	Indirect	Store	ea = (a)
4	Indexed, Postincrement	Store	ea = a + (r_a) $r_a \leftarrow r_a + 1$
5	Indexed, Predecrement	Store	$r_a \leftarrow r_a - 1$ ea = a + (r_a)
6	Indexed	Store	ea = a + (r_a)
7	Indexed, Indirect	Store	ea = (a + (r_a))

Mode 1 - Immediate. For fetch-type instructions, the content of the ADDRESS field is sign-extended and used as the operand. For store-type instructions, the immediate mode is the same as the direct mode (mode 2).

Mode 2 - Direct. The ADDRESS field of the instruction contains the memory address which is the source of the operand for fetch-type instructions or the destination address for store-type instructions.

A.3.2 Indirect Modes

Mode 3 - Indirect mode. The ADDRESS field of the instruction contains the memory address which contains the source/destination address for the instruction.

Mode 7 - Indexed, indirect. (See indexed modes)

A.3.3 Indexed Modes

Mode 4 - Indexed with (r_a) postincremented. The content of the ADDRESS field of the instruction is added to the content of r_a to produce the source/destination address for the instruction. After the address is computed, the content of r_a is incremented by one.

Mode 5 - Indexed with (r_a) predecremented. The content of r_a is decremented by one and this value is added to the content of the ADDRESS field to produce the source/destination address for the instruction.

Mode 6 - Indexed with (r_a). The content of the ADDRESS field is added to the content of r_a to produce the source/destination address for the instruction.

Mode 7 - Indexed, indirect. The content of the ADDRESS field is added to the content of r_a to produce the memory address which contains the source/destination address for the instruction.

A.4 MACHINE INSTRUCTION REPERTOIRE

The set of BFTSC instruction mnemonics along with their definitions and op codes is listed in this section.

A.4.1 Fetch-Type Instructions

A.4.1.1 Load Instructions

Op Code	Mnemonic	Definition
00	LDR	Load Register
01	LDE	Load Extension Register
02	LDR2	Load Multiple
03	LDR3	
04	LDR7	
2E	LDS2	Load Constant Multiple
2F	LDS3	
05	LDN	Load Negative Integer
06	LDNF	Load Negative Floating
07	LDA	Load Absolute Value Integer
08	LDAF	Load Absolute Value Floating
09	LDC	Load One's Complement
0A	LAO	Load Active Only
0B	LMO	Load Monitor Only
30	LW0	Load Working Reg. 0
31	LW1	Load Working Reg. 1
32	LW2	Load Working Reg. 2
33	LW3	Load Working Reg. 3

A.4.1.2 Floating-Point Arithmetic Instructions

Op Code	Mnemonic	Definition
0C	ADDF	Add Floating
0D	SUBF	Subtract Floating
0E	MPYF	Multiply Floating
0F	DIVF	Divide Floating

Op Code	Mnemonic	Definition
10	SRTF	Square Root Floating
11	VADDF	Vector Add Floating
12	VSUBF	Vector Subtract Floating
13	VMPYF	Vector Multiply Floating
14	VIPF	Vector Inner Product Floating
15	VSMF	Vector-Scalar Multiply Floating
16	CFX	Convert Floating to Integer
17	UPF	Unpack Floating
18	PKF	Pack Floating

A.4.1.3 Fixed-Point Integer Arithmetic Instructions

Op Code	Mnemonic	Description
19	ADD	Add Integer
2B	ACO	Add Integer with Carryout
1A	SUB	Subtract Integer
1B	MPY	Multiply Integer
1C	DIV	Short Dividend Divide Integer
1D	LDV	Long Dividend Divide Integer
1E	CFL	Convert Integer to Floating

A.4.1.4 Logical Instructions

Op Code	Mnemonic	Description
1F	AND	Logical AND
20	XOR	Logical Exclusive OR
21	IOR	Logical Inclusive OR
22	ANI	Logical AND Inverted

A.4.1.5 Shift/Rotate Instructions

Op Code	Mnemonic	Description
23	ARS	Arithmetic Short Shift
24	ARL	Arithmetic Long Shift
25	RRS	Rotate Short
26	RRL	Rotate Long
27	LRS	Logical Short Shift
28	LRL	Logical Long Shift

A.4.1.6 Test and Skip Instructions

Op Code	Mnemonic	Description
29	OISN	Logical OR Inverted and Skip if Not Ones
2D	OISO	Logical OR Inverted and Skip if Ones
2A	ASNZ	Logical AND and Skip if Not Zeros
2C	ASZ	Logical AND and Skip if Zeros
34	CSNE	Logical COMPARE and Skip if Not Equal
35	CSEQ	Logical COMPARE and Skip if Equal

A.4.2 Store-Type Instructions

A.4.2.1 Store Instructions

Op Code	Mnemonic	Description
40	STR	Store Register
41	STE	Store Extension Register
42	STD	Store Duplex
47	STZ	Store Zero
43	SZD	Store Zero Duplex
62	STR2	Store Multiple Single
63	STR3	

Op Code	Mnemonic	Description
44	STD2	Store Multiple Duplex
45	STD3	
46	STD7	
49	STH	Store to Hard Address
48	SPS	Store Program Counter and Status Single
4A	SPC	Store Program Counter and Status Duplex
4B	SBPA1	Store with Bad Address Parity
4C	SBPA0	
4D	SBPD1	Store with Bad Data Parity
4E	SBPD0	
65	SW0	Store Working Register 0
66	SW1	Store Working Register 1
67	SW2	Store Working Register 2
68	SW3	Store Working Register 3

A.4.2.2 Jump Instructions

Op Code	Mnemonic	Description
50	JMP	Jump
4F	JPZ	Jump if Positive or Zero Integer
51	JMI	Jump if Negative Integer
52	JZE	Jump if Zero Integer
53	JZEF	Jump if Zero Floating
54	JNZ	Jump if Not Zero Integer
55	JNZF	Jump if Not Zero Floating
56	JPS	Jump if Positive and Not Zero Integer
57	JPSF	Jump if Positive and Not Zero Floating
58	JMZ	Jump if Negative or Zero Integer
59	JMZF	Jump if Negative or Zero Floating

Op Code	Mnemonic	Description
5A	JDN	Decrement r_b Jump if Not Zero Integer
5C	JOS	Jump if Overflow Set and Reset Overflow
5D	JCS	Jump if Carryout Set and Reset Out
5E	JSB	Jump Subroutine

A.4.2.3 Miscellaneous

Op Code	Mnemonic	Description
5B	DSI	Disable Interrupts
5F	ENI	Enable Interrupts
60	RFI	Return from Interrupt
64	RET	Return from Subroutine
61	XEC	Execute

A.5 MACHINE INSTRUCTIONS

This section contains a description of the operation of each of the BFTSC instructions and the inputs and outputs required.

LOAD REGISTER — LDR (00)

NORMAL ADDRESS MODES — 0-7

MACHINE OPERATION

$(r_b) \leftarrow (ea)$

The contents of the effective address are loaded into the specified register.

LOAD EXTENSION REGISTER — LDE (01)

NORMAL ADDRESS MODES — 0-7

MACHINE OPERATION

$(ex) \leftarrow (ea)$

The contents of the effective address are loaded into the extension register.

NOTES

RB is unused in this instruction.

LOAD MULTIPLE

LDR2 (02)
LDR3 (03)
LDR7 (04)

NORMAL ADDRESS MODES 0, 2-7

MACHINE OPERATION

$(r_b) \leftarrow (ea)$ } LDR2 } LDR3 } LDR7
$(r_{b+1}) \leftarrow (ea + 1)$
$(r_{b+2}) \leftarrow (ea + 2)$

.
.
.

$(r_{b+6}) \leftarrow (ea + 6)$

Two, three or seven data words are loaded into 2, 3 or 7 successive registers starting with register r_b (r_0 follows r_7).

NOTES

When immediate mode (mode 1) is specified, the sign-extended "a" is loaded into the first register. The remaining registers are loaded as in direct mode (mode 2) where ea + 1 = a + 1, ea + 2 = a + 2, etc.

LOAD CONSTANT MULTIPLE — LDS2 (2E), LDS3 (2F)

NORMAL ADDRESS MODES — 0-7

MACHINE OPERATION

$(r_b) \leftarrow (ea)$ } LDS2
$(r_{b+1}) \leftarrow (ea)$ } LDS3
$(r_{b+2}) \leftarrow (ea)$ } LDS3
$(r_{b+3}) \leftarrow (ea)$ } LDS7
$(r_{b+4}) \leftarrow (ea)$ } LDS7
$(r_{b+5}) \leftarrow (ea)$ } LDS7
$(r_{b+6}) \leftarrow (ea)$ } LDS7

A constant is loaded into 2, 3, or 7 successive registers starting with register r_b (r_0 follows r_7).

LOAD NEGATIVE INTEGER — LDN (05)

NORMAL ADDRESS MODES — 0-7

MACHINE OPERATION

$(r_b) \leftarrow -(ea)$

The contents of the effective address are two's-complemented.

ERROR INDICATORS

Overflow status is set if $(ea) = -2^{31}$

LOAD NEGATIVE FLOATING — LDNF (06)

NORMAL ADDRESS MODES — 0, 2-7

MACHINE OPERATION

$(r_b) \leftarrow -(ea)$

The mantissa is two's-complemented and the number is normalized.

ERROR INDICATORS

Overflow status is set if the exponent overflows or underflows during normalization.

NOTES

When immediate mode (mode 1) is specified, the sign-extended "a" is used as the 32-bit, floating-point number to be complemented. This is an unnormalized floating-point number.

LOAD ABSOLUTE VALUE INTEGER — LDA (07)

NORMAL ADDRESS MODES — 0-7

MACHINE OPERATION

$(r_b) \leftarrow |(ea)|$

If the value of the data is < 0 (sign bit set) the data is two's-complemented and transferred to r_b.

ERROR INDICATORS

Overflow status is set if (ea) = -2^{31}.

LOAD ABSOLUTE VALUE FLOATING — LDAF (08)

NORMAL ADDRESS MODES — 0, 2-7

MACHINE OPERATION

$(r_b) \leftarrow |(ea)|$

If the mantissa of the effective address is greater than zero, the data is unaffected. If the mantissa is less than zero, it is two's-complemented and then normalized.

ERROR INDICATORS

Overflow status is set if the exponent overflows or underflows during normalization.

NOTES

When immediate mode (mode 1) is specified, the sign-extended "a" is used as the 32-bit, floating-point operand. This is an unnormalized floating-point number.

LOAD ONE'S COMPLEMENT — LDC (09)

NORMAL ADDRESS MODES — 0-7

MACHINE OPERATION

$(r_b) \leftarrow \overline{(ea)}$

The contents of the effective address are one's-complemented and loaded into the specified register.

LOAD ACTIVE ONLY — LAO (0A)

NORMAL ADDRESS MODES — 0-7

MACHINE OPERATION

$(r_b) \leftarrow (ea)$

In the active CPU only; the monitor CPU executes this instruction as an NOP instruction.

LOAD MONITOR ONLY — LMO (0B)

NORMAL ADDRESS MODES — 0-7

MACHINE OPERATION

$(r_b) \leftarrow (ea)$

In the monitor CPU only; the active CPU executes this instruction as an NOP instruction.

LOAD WORKING REGISTER — LW0 (30)
LW1 (31)
LW2 (32)
LW3 (33)

NORMAL ADDRESS MODES — 0-7

MACHINE OPERATION

(W0), (W1), (W2), or (W3) $\leftarrow$ (ea)

The contents of the effective address are loaded into the specified working register.

NOTES

RB is unused in this instruction.

ADD FLOATING — ADDF (0C)

NORMAL ADDRESS MODES — 0, 2-7

MACHINE OPERATOR

$(r_b) \leftarrow (r_b) + (ea)$

The floating-point content of the effective address is added to the floating-point content of register r_b and the result is placed in r_b. Normalized floating-point results are generated.

ERROR INDICATORS

Exponent overflow or underflow sets overflow status. Mantissa overflow is internally corrected and the exponent adjusted.

NOTES

When immediate mode (mode 1) is specified, the sign-extended "a" is used as the 32-bit, floating-point operand. This is an unnormalized number.

SUBTRACT FLOATING — SUBF (0D)

NORMAL ADDRESS MODES — 0, 2-7

MACHINE OPERATION

$(r_b) \leftarrow (r_b) - (ea)$

The floating-point content of the effective address is subtracted from the floating-point content of register r_b. The result is placed in r_b. Normalized floating-point results are generated.

ERROR INDICATORS

Exponent underflow or overflow sets overflow status. Mantissa overflow or underflow is internally corrected.

NOTES

When immediate mode (mode 1) is specified, the sign-extended "a" is used as the 32-bit, floating-point operand. This is an *unnormalized number.*

MULTIPLY FLOATING — MPYF (0E)

NORMAL ADDRESS MODES — 0, 2-7

MACHINE OPERATION

$(r_b) \leftarrow$ 24 MSB's of Mantissa and 8-bit exponent of (r_b) x (ea).
(ex) $\leftarrow$ 23 LSB's of Mantissa of product in most significant 23 bits followed by zeros.

The floating-point content of r_b is multiplied by the floating-point content of the effective address. The 24 MSB's of the resultant mantissa and the 8-bit exponent are placed in r_b. The 23 LSB's of the mantissa followed by zeros are placed in ex. Normalized floating-point results are generated.

ERROR INDICATORS

Exponent underflow or overflow sets overflow status. Mantissa overflow or underflow is internally corrected.

NOTES

When immediate mode (mode 1) is specified, the sign-extended "a" is used as the 32-bit, floating-point operand. This is an unnormalized number.

DIVIDE FLOATING — DIVF (0F)

NORMAL ADDRESS MODES — 0, 2-7

MACHINE OPERATION

$(r_b) \leftarrow (r_b) \div (ea)$ with mantissa truncated after 24 bits.

(ex) ← Remainder of the division. Contains the remainder sign bit followed by the 23-bit remainder followed by zeros.

The floating-point content of r_b is divided by the floating-point content of the effective address. The quotient (mantissa truncated after 24 bits) is placed in r_b and the remainder is placed in ex. Normalized floating-point results are generated.

ERROR INDICATORS

Exponent underflow or overflow sets overflow status. Divide by zero sets divide check status.

NOTES

When immediate mode (mode 1) is specified, the sign-extended "a" is used as the 32-bit, floating-point operand. This is an unnormalized number.

SQUARE ROOT FLOATING — SRTF (10)

NORMAL ADDRESS MODES — 0, 2-7

MACHINE OPERATION

$(r_b) \leftarrow$ SQRT (ea) with mantissa truncated after 24 bits.

The floating-point square root of the floating-point content of ea is placed in register r_b. Normalized floating-point results are generated.

ERROR INDICATORS

A negative operand sets overflow status.

NOTES

When immediate mode (mode 1) is specified, the sign-extended "a" is used as the 32-bit, floating-point operand. This is an unnormalized number and will produce incorrect results.

VECTOR ADD FLOATING — VADDF (11)

NORMAL ADDRESS MODES — 0, 2-7

MACHINE OPERATION

$(r_b) \leftarrow (r_b) + (ea)$

$(r_{b+1}) \leftarrow (r_{b+1}) + (ea + 1)$

$(r_{b+2}) \leftarrow (r_{b+2}) + (ea + 2)$

The length three vector of operands is added to the three specified registers (r_0 follows r_7). Normalized floating-point results are generated.

ERROR INDICATORS

Exponent overflow or underflow sets overflow status.

NOTES

When immediate mode (mode 1) is specified, the sign-extended "a" is used as the first operand. This is an unnormalized floating-point number. The second and third operands are fetched as in direct mode (mode 2) from a + 1 and a + 2.

VECTOR SUBTRACT FLOATING — VSUBF (12)

NORMAL ADDRESS MODES — 0, 2-7

MACHINE OPERATION

$(r_b) \leftarrow (r_b) - (ea)$

$(r_{b+1}) \leftarrow (r_{b+1}) - (ea + 1)$

$(r_{b+2}) \leftarrow (r_{b+2}) - (ea + 2)$

The length three vector of operands is subtracted from the three specified registers (r_0 follows r_7). Normalized floating-point results are generated.

ERROR INDICATORS

Exponent overflow or underflow sets overflow status.

NOTES

When immediate mode (mode 1) is specified, the sign-extended "a" is used as the first operand. This is an unnormalized floating-point number. The second and third operands are fetched as in direct mode (mode 2) from a + 1 and a + 2.

VECTOR MULTIPLY FLOATING

VMPYF (13)

NORMAL ADDRESS MODES

0, 2-7

MACHINE OPERATION

$(r_b) \leftarrow (r_b) \times (ea)$

$(r_{b+1}) \leftarrow (r_{b+1}) \times (ea + 1)$

$(r_{b+2}) \leftarrow (r_{b+2}) \times (ea + 2)$

The length three vector operand is element by element multiplied by the specified registers (r_0 follows r_7). Normalized floating-point results are generated.

ERROR INDICATORS

Exponent overflow or underflow sets overflow status.

NOTES

When immediate mode (mode 1) is specified, the sign-extended "a" is used as the first operand. This is an unnormalized floating-point number. The second and third operands are fetched as in direct mode (mode 2) from a + 1 and a + 2.

In the first two multiplies the 23 LSBs of the mantissa are destroyed. In the third multiply the 23 LSBs are contained in the most significant 23 bits of the extension register followed by all zeros.

VECTOR INNER PRODUCT FLOATING (DOT PRODUCT) VIPF (14)

NORMAL ADDRESS MODES 0, 2-7

MACHINE OPERATION

$(r_b) \leftarrow (r_b) \times (ea)$

$(r_{b+1}) \leftarrow (r_{b+1}) \times (ea + 1)$

$(r_{b+2}) \leftarrow (r_{b+2}) \times (ea + 2)$

$(r_{b+2}) \leftarrow (r_b) + (r_{b+1}) + (r_{b+2})$

The length three vector operand is element by element multiplied by the specified registers (r_0 follows r_7), and the sum of the products is put in the last register (r_{b+2}). Normalized floating-point results are generated.

ERROR INDICATORS

An exponent overflow or underflow in any multiply or add sets overflow status.

NOTES

When immediate mode (mode 1) is specified, the sign-extended "a" is used as the first operand. This is an unnormalized floating-point number. The second and third operands are fetched as in direct mode (mode 2) from a + 1 and a + 2.

In the first two multiplies the 23 LSBs of the mantissa are destroyed. In the third multiply the 23 LSBs are contained in the most significant 23 bits of the extension register followed by all zeros.

VECTOR-SCALAR MULTIPLY FLOATING VSMF (15)

NORMAL ADDRESS MODES 0, 2-7

MACHINE OPERATION

$(r_b) \leftarrow (r_b) \times (ea)$

$(r_{b+1}) \leftarrow (r_{b+1}) \times (ea)$

$(r_{b+2}) \leftarrow (r_{b+2}) \times (ea)$

The three register vector components (r_0 follows r_7) are multiplied by the floating-point scalar content of the ea. Normalized floating-point results are generated.

ERROR INDICATORS

Exponent overflow or underflow sets overflow status.

NOTES

When immediate mode (mode 1) is specified, the sign-extended "a" is used as the 32-bit, scalar operand. This is an unnormalized floating-point number.

In the first two multiplies the 23 LSBs of the mantissa are destroyed. In the third multiply the 23 LSBs are contained in the most significant 23 bits of the extension register followed by all zeros.

CONVERT FLOATING TO INTEGER — CFX (16)

NORMAL ADDRESS MODES — 0, 2-7

MACHINE OPERATION

$(r_b) \leftarrow$ (ea) converted to integer format

The mantissa of the content of the ea is shifted while the exponent is adjusted until the exponent equals 31. The fractional part of the number is truncated.

ERROR INDICATORS

Conversion of numbers $>2^{31}-1$ or $<-2^{31}$ sets overflow status.

NOTES

When immediate mode (mode 1) is specified, the sign-extended "a" is used as the 32-bit, floating-point operand. This is an unnormalized number.

UNPACK FLOATING — UPF (17)

NORMAL ADDRESS MODES — 0, 2-7

MACHINE OPERATION

$(r_b) \leftarrow$ Exponent of (ea) (right justified with sign extension)

$(r_{b+1}) \leftarrow$ Mantissa cf (ea) (left justified with trailing zeros)

The floating-point content of the effective address is unpacked into a mantissa word and an exponent word (r_0 follows r_7). The 24-bit mantissa is left justified with trailing zeros and the exponent is right justified and sign extended.

NOTES

When immediate mode (mode 1) is specified, the sign-extended "a" is used as the 32-bit, floating-point operand. This is an unnormalized number.

PACK FLOATING — PKF (18)

NORMAL ADDRESS MODES — 0, 2-7

MACHINE OPERATION

$(r_b) \leftarrow$ 24 MSB's of normalized mantissa concatenated with 8 LSB's of adjusted exponent.

The normalized mantissa and adjusted exponent are packed into r_b.

For mode 0 (register to register mode), the exponent is (r_b) and the mantissa to be normalized is (r_a). For modes 2-7, the exponent is (ea) and the mantissa is (ea + 1).

ERROR INDICATORS

Exponent underflow sets overflow status.

NOTES

When immediate mode (mode 1) is specified, the sign-extended "a" is used as the exponent operand. The mantissa to be normalized is fetched as in direct mode (mode 2) from a + 1.

ADD INTEGER — ADD (19)

NORMAL ADDRESS MODES — 0-7

MACHINE OPERATION

$(r_b) \leftarrow (r_b) + (ea)$

The integer content of the effective address is added to the integer content of register r_b, and the result is put into r_b.

ERROR INDICATORS

Two's-complement overflow or underflow (reversed sign) sets overflow status.

Two's-complement carryout sets carryout status.

ADD INTEGER WITH CARRYOUT — ACO (2B)

NORMAL ADDRESS MODES — 0-7

MACHINE OPERATION

$(r_b) \leftarrow (r_b) + (ea) + (CO)$

$(CO) \leftarrow$ new carryout

The integer content of the effective address, the integer content of register r_b, and the content of the carryout status flop (0 or 1) are added. The integer result is put into r_b and the resultant carryout is put into the carryout status flop.

ERROR INDICATORS

A two's-complement overflow or underflow sets overflow status.

SUBTRACT INTEGER SUB (1A)

NORMAL ADDRESS MODES 0-7

MACHINE OPERATION

$(r_b) \leftarrow (r_b) - (ea)$

The integer content of the effective address is subtracted from the integer content of register r_b. The result is placed in r_b.

ERROR INDICATORS

Two's-complement overflow or underflow (reversed sign) sets overflow status.

Two's-complement carryout sets carryout status.

MULTIPLY INTEGER MPY (1B)

NORMAL ADDRESS MODES 0-7

MACHINE OPERATION

$(r_b) \leftarrow$ 32 MSB's of (r_b) x (ea).

(ex) $\leftarrow$ Sign bit of the product followed by 31 LSB's of the product.

The integer content of register r_b is multiplied by the integer content of the effective address. The 63-bit, integer result is packed with the sign bit and the 31 LSB's in r_b and the 32 MSB's in ex.

ERROR INDICATORS

When -2^{31} (80000000 hex) is multiplied by itself, overflow status is set.

SHORT DIVIDEND DIVIDE INTEGER — DIV (1C)

LONG DIVIDEND DIVIDE INTEGER — LDV (1D)

NORMAL ADDRESS MODES — 0-7

MACHINE OPERATION

For DIV:

$(r_b) \leftarrow (r_b) \div (ea)$

(ex) ← Remainder of the division.

The integer content of register r_b is divided by the integer content of the effective address. The integer quotient is placed in r_b and the remainder is placed in ex.

For LDV:

$(r_b) \leftarrow \left[(ex), (r_b)\right] \div (ea)$

(ex) ← Remainder of the division.

The 63-bit, integer dividend, produced by concatenating the 32-bit integer content of the extension register with the 31 LSB's of register r_b, is divided by the 32-bit, integer content of the effective address. The 32-bit integer quotient is placed in r_b and the 32-bit, integer remainder is placed in ex.

ERROR INDICATORS

Divide by zero sets divide check status.

Overflow status is set when the quotient requires more than 32 bits including sign.

CONVERT INTEGER TO FLOATING — CFL (1E)

NORMAL ADDRESS MODES — 0-7

MACHINE OPERATION

$(r_b) \leftarrow$ (ea) converted to floating-point format.

The integer content of the effective address is normalized to form the mantissa while the corrected exponent is formed. The result, consisting of the 24-bit, truncated mantissa concatenated with the 8-bit exponent, is placed in r_b.

LOGICAL AND — AND (1F)

NORMAL ADDRESS MODES — 0-7

MACHINE OPERATION

$(r_b) \leftarrow (r_b) \wedge$ (ea)

The contents of register r_b are ANDed with the contents of the effective address and the result placed in r_b.

LOGICAL EXCLUSIVE OR — XOR (20)

NORMAL ADDRESS MODES — 0-7

MACHINE OPERATION

$(r_a) \leftarrow (r_b) \oplus$ (ea)

The contents of r_b are EXCLUSIVE ORed with the contents of the effective address and the result placed in r_b.

LOGICAL INCLUSIVE OR — IOR (21)

NORMAL ADDRESS MODES — 0-7

MACHINE OPERATION

$(r_b) \leftarrow (r_b) \vee (ea)$

The contents of register r_b are INCLUSIVE ORed with the contents of the effective address and the result placed in r_b.

LOGICAL AND INVERTED — ANI (22)

NORMAL ADDRESS MODES — 0-7

MACHINE OPERATION

$(r_b) \leftarrow (r_b) \wedge \overline{(ea)}$

The contents of register r_b are ANDed with the inverted contents of the effective address. The result is placed in r_b.

ARITHMETIC SHORT SHIFT — ARS (23)

NORMAL ADDRESS MODES — 0-7

MACHINE OPERATION

$(r_b) \leftarrow$ shifted (r_b); left if (ea) < 0; right if (ea) > 0

The content of register r_b is shifted the amount specified by the least significant 8 bits of the content of ea. A right shift causes sign extension, a left shift shifts in zeros.

ERROR INDICATORS

If the sign bit changes during shifting, overflow status is set.

NOTES

The two's-complement shift count contained in the lower 8 bits of (ea) is used (the sign bit is bit 24).

ARITHMETIC LONG SHIFT — ARL (24)

NORMAL ADDRESS MODES — 0-7

MACHINE OPERATION

(r_b), (ex) $\leftarrow$ shifted $\left[(r_b, (ex)\right]$; left if (ea) < 0; right if (ea) > 0

The contents of the register r_b concatenated with the lower 31 bits of (ex) are shifted the amount specified by the least significant 8 bits of the content of ea. A right shift causes sign extension, a left shift shifts in zeros. The MSB of the extension register (sign bit) is set to 0.

ERROR INDICATORS

If the sign bit of (r_b) changes during shifting, overflow status is set.

NOTES

The two's-complement shift count contained in the lower 8 bits of (ea) is used (the sign bit is bit 24).

ROTATE SHORT — RRS (25)

NORMAL ADDRESS MODES — 0-7

MACHINE OPERATION

$(r_b) \leftarrow$ rotated (r_b); left if (ea) < 0; right if (ea) > 0

The content of register r_b is rotated the amount specified by the least significant 8 bits of the content of ea.

NOTES

The two's-complement shift count contained in the lower 8 bits of (ea) is used (the sign bit is bit 24).

ROTATE LONG — RRL (26)

NORMAL ADDRESS MODES — 0-7

MACHINE OPERATION

(r_b), (ex) $\leftarrow$ rotated $\left[(r_b),\ (ex)\right]$; left if (ea) < 0; right if (ea) > 0

The content of register r_b concatenated with the content of the extension register are rotated the amount specified by the least significant 8 bits of the content of ea.

NOTES

The two's-complement shift count contained in the lower 8 bits of (ea) is used (the sign bit is bit 24).

LOGICAL SHORT SHIFT — LRS (27)

NORMAL ADDRESS MODES — 0-7

MACHINE OPERATION

$(r_b) \leftarrow$ shifted (r_b); left if ea < 0; right if $(ea) > 0$

The content of register r_b is shifted the amount specified by the least significant 8 bits of the content of ea. Zeros are shifted in.

NOTES

The two's-complement shift count contained in the lower 8 bits of (ea) is used (the sign bit is bit 24).

LOGICAL LONG SHIFT — LRL (28)

NORMAL ADDRESS MODES — 0-7

MACHINE OPERATION

$(r_b),\ (ex) \leftarrow \text{shifted} \left[(r_b),\ (ex) \right]$; left if $(ea) < 0$; right if $(ea) > 0$

The content of register r_b concatenated with the content of the extension register is shifted the amount specified by the least significant 8 bits of the content of ea. Zeros are shifted in.

NOTES

The two's-complement shift count contained in the lower 8 bits of (ea) is used (the sign bit is bit 24).

LOGICAL OR INVERTED AND SKIP IF NOT ONES — OISN (29)

NORMAL ADDRESS MODES — 0-7

MACHINE OPERATION

If $\overline{(ea)} \vee (r_b) \neq$ 1's; (PC) ← (PC) + 1

A logical INCLUSIVE OR is performed on the inversion of (ea) and (r_b) (not altering either operand). If the result is not all 1's, the content of PC is incremented causing an instruction to be skipped.

LOGICAL OR INVERTED AND SKIP IF ONES — OISO (2D)

NORMAL ADDRESS MODES — 0-7

MACHINE OPERATION

If $\overline{(ea)} \vee (r_b) =$ 1's; (PC) ← (PC) + 1

A logical INCLUSIVE OR is performed on the inversion of (ea) and (r_b) (not altering either operand). If the result is all 1's, the content of PC is incremented causing an instruction to be skipped.

LOGICAL AND AND SKIP IF NOT ZEROS — ASNZ (2A)

NORMAL ADDRESS MODES — 0-7

MACHINE OPERATION

If (ea) $\wedge$ (r_b) $\neq$ 0's; (PC) $\leftarrow$ (PC) + 1

A logical AND is performed on (ea) and (r_b) (not altering either operand). If the result is not all O's, the content of PC is incremented causing an instruction to be skipped.

LOGICAL AND AND SKIP IF ZEROS — ASZ (2C)

NORMAL ADDRESS MODES — 0-7

MACHINE OPERATION

If (ea) $\wedge$ (r_b) = 0's; (PC) $\leftarrow$ (PC) + 1

A logical AND is performed on (ea) and (r_b) (not altering either operand). If the result is all 0's, the content of PC is incremented causing an instruction to be skipped.

LOGICAL COMPARE AND SKIP OF NOT EQUAL — CSNE (34)

NORMAL ADDRESS MODES — 0-7

MACHINE OPERATION

If (ea) $\neq$ (r_b); (PC) $\leftarrow$ (PC) + 1

A logical EXCLUSIVE OR is performed on (ea) and (r_b) (not altering either operand). If the result is not all 0's, the content of PC is incremented causing an instruction to be skipped.

LOGICAL COMPARE AND SKIP IF EQUAL — CSEQ (35)

NORMAL ADDRESS MODES — 0-7

MACHINE OPERATION

If (ea) = (r_b); (PC) $\leftarrow$ (PC) + 1

A logical EXCLUSIVE OR is performed on (ea) and (r_b) (not altering either operand). If the result is all 0's, the content of PC is incremented causing an instruction to be skipped.

STORE REGISTER — STR (40)

NORMAL ADDRESS MODES — 0, 2-7

MACHINE OPERATION

$(ea) \leftarrow (r_b)$

The contents of the specified register are stored into the effective address.

NOTES

When immediate mode (mode 1) is specified, the instruction is executed as though direct mode (mode 2) were specified.

STORE EXTENSION REGISTER — STE (41)

NORMAL ADDRESS MODES — 0, 2-7

MACHINE OPERATION

$(ea) \leftarrow (ex)$

The contents of the extension register are stored into the effective address.

NOTES

When immediate mode (mode 1) is specified, the instruction is executed as though direct mode (mode 2) were specified.

RB is unused in this instruction.

STORE DUPLEX — STD (42)

NORMAL ADDRESS MODES — 2-7

MACHINE OPERATION

$(ea) \leftarrow (r_b)$

$(ea + 4096) \leftarrow (r_b)$

The contents of the specified register are stored into ea in one memory block and also into ea + 4096 in the following memory block.

NOTES

When register to register mode (mode 0) is specified, an NOP instruction is performed.

When immediate mode (mode 1) is specified, the instruction is executed as though direct mode (mode 2) were specified.

STORE ZERO — STZ (47)

NORMAL ADDRESS MODES — 0, 2-7

MACHINE OPERATION

$(ea) \leftarrow 0$'s

All zeros are stored into the effective address.

NOTES

When immediate mode (mode 1) is specified, the instruction is executed as though direct mode (mode 2) were specified.

RB is not used in this instruction.

STORE ZERO DUPLEX — SZD (43)

NORMAL ADDRESS MODES — 2-7

MACHINE OPERATION

(ea) ← 0's

(ea + 4096) ← 0's

All zeros are stored into ea in one memory block and also into ea + 4096 in the following memory block.

NOTES

When register to register mode (mode 0) is specified, an NOP instruction is performed.

When immediate mode (mode 1) is specified, the instruction is executed as though direct mode (mode 2) were specified.

RB is not used in this instruction.

STORE MULTIPLE SINGLE

STR2 (62)
STR3 (63)

NORMAL ADDRESS MODES

0, 2-7

MACHINE OPERATION

$$\left.\begin{array}{l}(ea) \leftarrow (r_b) \\ (ea+1) \leftarrow (r_{b+1})\end{array}\right\} \text{STR2} \quad \left.\begin{array}{l}\\ \\ (ea+2) \leftarrow (r_{b+2})\end{array}\right\} \text{STR3}$$

The contents of 2 or 3 registers starting with r_b (r_0 follows r_7) are stored into 2 or 3 successive addresses starting with ea.

NOTES

When immediate mode (mode 1) is specified, the instruction is executed as though direct mode (mode 2) were specified.

STORE MULTIPLE DUPLEX

STD2 (44)
STD3 (45)
STD7 (46)

NORMAL ADDRESS MODES 0, 2-7

MACHINE OPERATION

$(ea) \leftarrow (r_b)$
$(ea + 4096) \leftarrow (r_b)$ } STD2
$(ea + 1) \leftarrow (r_{b+1})$
$(ea + 4096 + 1) \leftarrow (r_{b+1})$ } STD3
$(ea + 2) \leftarrow (r_{b+2})$
$(ea + 4096 + 2) \leftarrow (r_{b+2})$ } STD7
.
.
.
$(ea + 6) \leftarrow (r_{b+6})$
$(ea + 4096 + 6) \leftarrow (r_{b+6})$

The contents of 2, 3 or 7 registers starting with r_b (r_0 follows r_7) are stored into 2, 3 or 7 successive pairs of memory addresses starting with ea in one memory block and ea + 4096 in the following memory block.

NOTES

When register to register mode (mode 0) is specified, an NOP instruction is performed.

When immediate mode (mode 1) is specified, the instruction is executed as though direct mode (mode 2) were specified.

STORE TO HARD ADDRESS — STH (49)

NORMAL ADDRESS MODES — 2-7

MACHINE OPERATION

The hard address control function specified by the effective address is performed.

This instruction stores to the effective address in the hard address spectrum by placing ea onto the address bus and onto the 16 MSBs of the data bus. The sign bit of ea (same as data bit 0) is placed onto the 16 LSBs of the data bus and the hard address control line is enabled.

NOTES

When register to register mode (mode 0) or immediate mode (mode 1) is specified, the instruction is executed as though direct mode (mode 2) were specified.

RB is not used in this instruction.

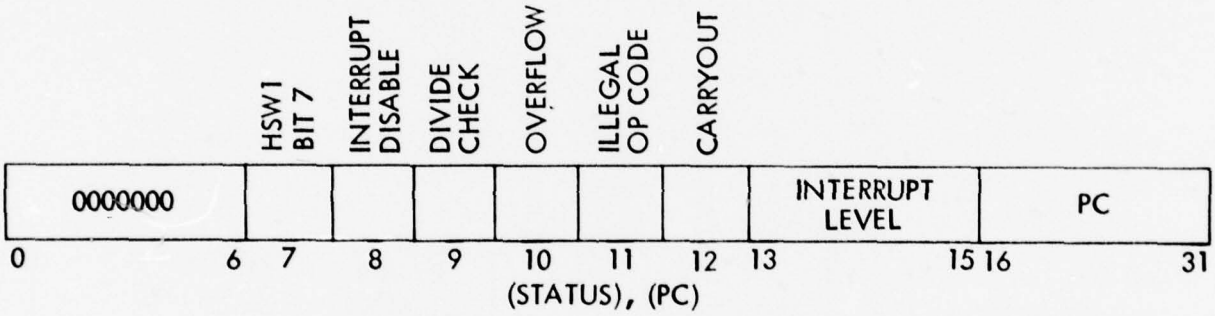
STORE PC & STATUS SINGLE — SPS (48)

NORMAL ADDRESS MODES — 0, 2-7

MACHINE OPERATION

(ea) ← (status), (PC)

The contents of the status register concatenated with the content of PC are stored into the effective address. The status register contains interrupt disable, divide check, overflow, illegal op code, and carryout status, and the level of the highest priority pending interrupt.



NOTES

When immediate mode (mode 1) is specified, the instruction is executed as though direct mode (mode 2) were specified.

RB is not used in this instruction.

Bit 7 is not used. It is present only so that the active and monitor CPUs have the information.

STORE PC & STATUS DUPLEX — SPC (4A)

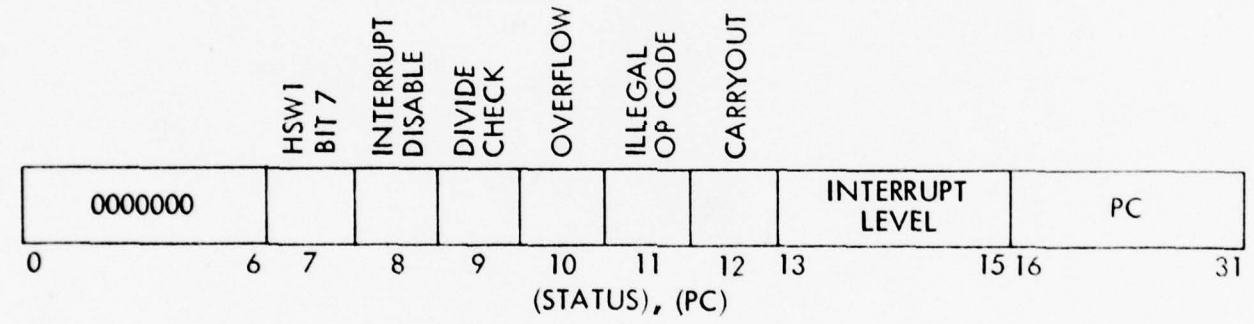
NORMAL ADDRESS MODES — 2-7

MACHINE OPERATION

(ea) ← (status), (PC)

(ea + 4096) ← (status), (PC)

The contents of the status register concatenated with the content of PC are stored into ea in one memory block and also into ea + 4096 in the following memory block. The status register contains interrupt disable, divide check, overflow, illegal op code, and carryout status, and the level of the highest priority pending interrupt.



NOTES

When register to register mode (mode 0) is specified, an NOP instruction is performed.

When immediate mode (mode 1) is specified, the instruction is executed as though direct mode (mode 2) were specified.

RB is not used in this instruction.

Bit 7 is not used. It is present only so that the active and monitor CPUs have the same information.

STORE WITH BAD ADDRESS PARITY — SBPA1 (4B)
SBPA0 (4C)

NORMAL ADDRESS MODES — 2-7

MACHINE OPERATION

$(ea) \leftarrow (r_b)$

The contents of the specified register are stored into the effective address. The 8 parity bits appended to the address on the address bus are, for these instructions, generated by the instruction as all ones (SBPA1) or all zeros (SBPA0) instead of the proper parity.

NOTES

When register to register mode (mode 0) is specified, a normal register to register store takes place; address parity is not generated.

When immediate mode (mode 1) is specified, the instruction is executed as though direct mode (mode 2) were specified.

STORE WITH BAD DATA PARITY

SBPD1 (4D)
SBPD0 (4E)

NORMAL ADDRESS MODES 2-7

MACHINE OPERATION

$(ea) \leftarrow (r_b)$

The contents of the specified register are stored into the effective address. The 8 parity bits appended to the data on the data bus are, for these instructions, generated by the instruction as all ones (SBPD1) or all zeros (SBPD0) instead of the proper parity.

NOTES

When register to register mode (mode 0) is specified, a normal register to registor store takes place; data parity is not generated.

When immediate mode (mode 1) is specified, the instruction is executed as though direct mode (mode 2) were specified.

STORE WORKING REGISTER — SW0 (65), SW1 (66), SW2 (67), SW3 (68)

NORMAL ADDRESS MODES — 0, 2-7

MACHINE OPERATION

(ea) ← (W0), (W1), (W2), or (W3)

The contents of the specified working register are stored into the effective address.

NOTES

When immediate mode (mode 1) is specified, the instruction is executed as though direct mode (mode 2) were specified.

RB is not used in this instruction.

JUMP — JMP (50)

NORMAL ADDRESS MODES — 0, 2-7

MACHINE OPERATION

(PC) ← ea for modes 2-7

(PC) ← (r_a) for mode 0

The content of PC is replaced by ea for modes 2-7, by (r_a) for mode 0.

NOTES

When immediate mode (mode 1) is specified, the instruction is executed as though direct mode (mode 2) were specified.

RB is not used in this instruction.

JUMP IF POSITIVE OR ZERO INTEGER — JPZ (4F)

NORMAL ADDRESS MODES — 0, 2-7

MACHINE OPERATION

If $(r_b) \geq 0$, (PC) ← ea for modes 2-7
(PC) ← (r_a) for mode 0

If the content of the specified register is positive or zero, the content of PC is replaced by ea for modes 2-7, by (r_a) for mode 0.

NOTES

When immediate mode (mode 1) is specified, the instruction is executed as though direct mode (mode 2) were specified.

JUMP IF NEGATIVE INTEGER — JMI (51)

NORMAL ADDRESS MODES — 0, 2-7

MACHINE OPERATION

If $(r_b) < 0$, (PC) ← ea for modes 2-7
(PC) ← (r_a) for mode 0

If the content of the specified register is negative, the content of PC is replaced by ea for modes 2-7, by (r_a) for mode 0.

NOTES

When immediate mode (mode 1) is specified, the instruction is executed as though direct mode (mode 2) were specified.

JUMP IF ZERO INTEGER — JZE (52)
JUMP IF ZERO FLOATING — JZEF (53)

NORMAL ADDRESS MODES — 0, 2-7

MACHINE OPERATION

If $(r_b) = 0$, (PC) ← ea for modes 2-7
(PC) ← (r_a) for mode 0

If the content of the specified register is zero, the content of PC is replaced by ea for modes 2-7, by (r_a) for mode 0.

NOTES

When immediate mode (mode 1) is specified, the instruction is executed as though direct mode (mode 2) were specified.

JUMP IF NOT ZERO INTEGER — JNZ (54)
JUMP IF NOT ZERO FLOATING — JNZF (55)

NORMAL ADDRESS MODES — 0, 2-7

MACHINE OPERATION

If $(r_b) \neq 0$, (PC) ← ea for modes 2-7
(PC) ← (r_a) for mode 0

If the content of the specified register is not zero, the content of PC is replaced by ea for modes 2-7, by (r_a) for mode 0.

NOTES

When immediate mode (mode 1) is specified, the instruction is executed as though direct mode (mode 2) were specified.

JUMP IF POSITIVE AND NOT ZERO INTEGER JPS (56)
JUMP IF POSITIVE AND NOT ZERO FLOATING JPSF (57)

NORMAL ADDRESS MODES 0, 2-7

MACHINE OPERATION

If $(r_b) > 0$, (PC) ← ea for modes 2-7
(PC) ← (r_a) for mode 0

If the content of the specified register is positive and not zero, the content of PC is replaced by ea for modes 2-7, by (r_a) for mode 0.

NOTES

When immediate mode (mode 1) is specified, the instruction is executed as though direct mode (mode 2) were specified.

JUMP IF NEGATIVE OR ZERO INTEGER JMZ (58)
JUMP IF NEGATIVE OR ZERO FLOATING JMZF (59)

NORMAL ADDRESS MODES 0, 2-7

MACHINE OPERATION

If $(r_b) \leq 0$, (PC) ← ea for modes 2-7
(PC) ← (r_a) for mode 0

If the content of the specified register is negative or zero, the content of PC is replaced by ea for modes 2-7, by (r_a) for mode 0.

NOTES

When immediate mode (mode 1) is specified, the instruction is executed as though direct mode (mode 2) were specified.

DECREMENT (r_b), JUMP IF NOT ZERO INTEGER — JDN (5A)

NORMAL ADDRESS MODES — 0, 2-7

MACHINE OPERATION

Initially $(r_b) \leftarrow (r_b) - 1$

Then if $(r_b) \neq 0$, $(PC) \leftarrow$ ea for modes 2-7

$(PC) \leftarrow (r_a)$ for mode 0

The content of the specified register is decremented by one. If the result is not zero, the content of PC is replaced by ea for modes 2-7, by (r_a) for mode 0.

NOTES

When immediate mode (mode 1) is specified, the instruction is executed as though direct mode (mode 2) were specified.

JUMP IF OVERFLOW SET AND RESET OVERFLOW — JOS (5C)

NORMAL ADDRESS MODES — 0, 2-7

MACHINE OPERATION

If (OV) = 1, $(PC) \leftarrow$ ea for modes 2-7

$(PC) \leftarrow (r_a)$ for mode 0

$(OV) \leftarrow 0$

If the overflow status flop is set, the content of PC is replaced by ea for modes 2-7, by (r_a) for mode 0. The overflow status flop is then reset.

NOTES

When immediate mode (mode 1) is specified, the instruction is executed as though direct mode (mode 2) were specified.

RB is not used in this instruction.

JUMP IF CARRYOUT SET AND RESET CARRYOUT — JCS (5D)

NORMAL ADDRESS MODES — 0, 2-7

MACHINE OPERATION

If (CO) = 1, (PC) ← ea for modes 2-7
(PC) ← (r_a) for mode 0

(CO) ← 0

If the carryout status flop is set, the content of PC is replaced by *ea for modes* 2-7, by (r_a) *for mode* 0. The carryout status flop is then reset.

NOTES

When immediate mode (mode 1) is specified, the instruction is executed as though direct mode (mode 2) were specified.

RB is not used in this instruction.

JUMP SUBROUTINE — JSB (5E)

NORMAL ADDRESS MODES — 0, 2-7

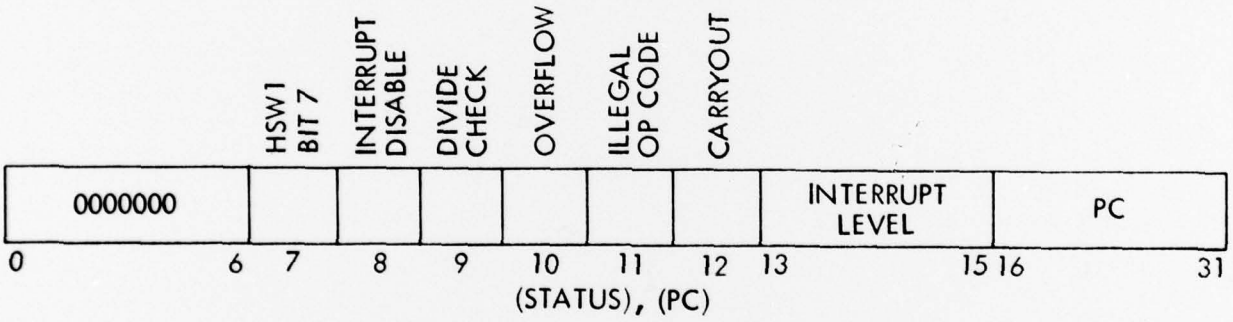
MACHINE OPERATION

$(r_b) \leftarrow$ (status), (PC)

(PC) $\leftarrow$ ea for modes 2-7

(PC) $\leftarrow$ (r_a) for mode 0

The contents of the status register concatenated with the content of PC are stored into the specified register. Then the content of PC is replaced by ea for modes 2-7, by (r_a) for mode 0. The status register contains interrupt disable, divide check, overflow, illegal OP CODE, and carryout status, and the level of the highest priority pending interrupt.



NOTES

When immediate mode (mode 1) is specified, the instruction is executed as though direct mode (mode 2) were specified.

Bit 7 is not used. It is present only so that the active and monitor CPUs have the same information.

DISABLE INTERRUPTS — DSI (5B)
ENABLE INTERRUPTS — ENI (5F)

NORMAL ADDRESS MODES — None Used

MACHINE OPERATION

Interrupts are disabled (DSI)

or

Interrupts are enabled (ENI)

NOTES

All fields except the OP CODE are unused.

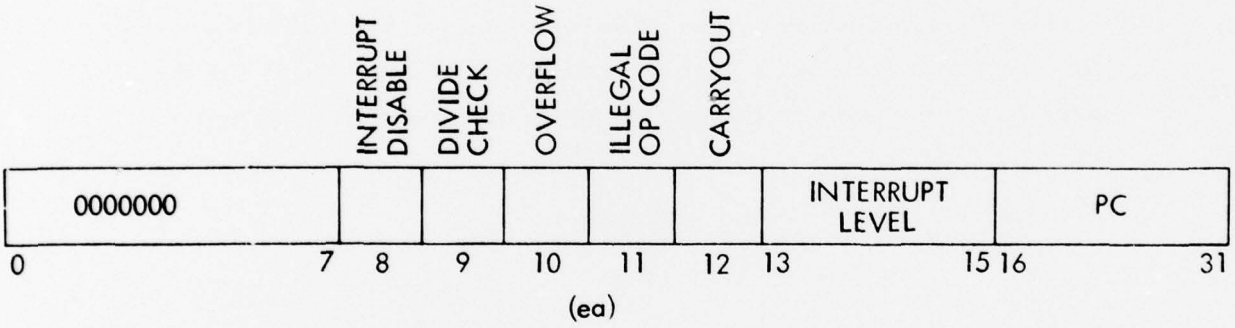
RETURN FROM INTERRUPT — RFI (60)
RETURN FROM SUBROUTINE — RET (64)

NORMAL ADDRESS MODES — 0, 2-7

MACHINE OPERATION

(status), (PC) ← (ea)

The contents of the interrupt disable, divide check, overflow, illegal OP CODE, and carryout status flops are replaced by (ea) bits 8-12. The content of PC is replaced by (ea) bits 16-31. For RFI, the in-process flop for the interrupt level specified by (ea) bits 13-15 is reset. For RET, the in-process flop is not reset.



NOTES

When immediate mode (mode 1) is specified, the instruction is executed as though direct mode (mode 2) were specified.

RB is not used in this instruction.

EXECUTE XEC (61)

NORMAL ADDRESS MODES 0, 2-7

MACHINE OPERATION

The instruction contained in the effective address is executed.

This instruction does not alter PC in order to fetch the instruction in ea. PC is altered only if the instruction in ea causes PC to change (i.e., jump).

NOTES

When immediate mode (mode 1) is specified, the instruction is executed as though direct mode (mode 2) were specified.

If an interrupt occurs before the content of ea is fetched, the EXECUTE instruction is restarted upon return from the interrupt. Any register modified by the housekeeping (mode 4 or 5) for the previous fetch (before the interrupt) of this instruction is not reinitialized.

RB is not used in this instruction.

APPENDIX B

GLOSSARY

Active CPU	- One of the two powered CPUs; provides its outputs to the other modules and controls the computer.
Alexic State	- State in which all volatile storage and configuration status information is assumed to be lost (e.g., following a power interruption).
Amnestic State	- State in which all information is assumed to be lost (e.g., cold start).
Bus Arbiter	- The portion of the CPU which exercises administrative control over access to the address, data, and control buses.
Bus State	- The specific set of three address-bus bytes and five data-bus bytes which are active at any given time.
CAT I, II, III, and IV Faults	- Category I, II, III, and IV Faults (see Fault Category).
Coverage	- The conditional probability that the system recovers from a fault given that sufficient operational hardware is available.
CCU Clamp	- A signal from each Circumvention Module to the CCU Modules indicating a radiation event or a power switchover.
Circumvention	- The inhibition of all I/O and memory operations during a radiation event or a power interruption.
CPU State	- The specific pair of CPUs assigned to the active and monitor CPU roles at any given time.
Dual Redundancy	- A pair of identical elements assigned to the same function, either one of which can assume the active role with the other serving as a spare.
Energy Storage	- A function of each Power Module; provides sufficient energy to power critical circuits during the time required for a power switchover to the standby Power Module or a radiation event.

Error Codes	- Two shortened cyclic codes used to protect information within the BFTSC. The bus code is used to protect all addresses and data passed among BFTSC modules and over the serial bus; it is defined by the generator polynomial $g(x) = (x^6 + x + 1)(x^2 + x + 1)$. The memory code is used to protect all information stored in main memory; it is defined by the generator polynomial $g'(x) = x^6 + x + 1$. (Same as Parity for FTSC.)
Error Correction	- A function of each Memory Module; when in the error correction mode, a Memory Module examines each word read and alters any single incorrect bit to produce a properly encoded word.
Fault Category	- Any one of the four categories into which faults in the BFTSC are classified; viz: power interruptions or radiation events (Category I), CPU faults (Category II), address or data bus faults (Category III), and I/O module, watchdog timer, or Timing Module faults (Category IV).
Fault Interrupt	- A signal from each CCU module indicating that a fault has been reported by one or more of the fault monitors within the BFTSC.
Fault Tolerance	- The ability to tolerate faults; i.e., to retain operational capability in the presence of failures.
Flags 1, 2, and 3	- Three discrete signals set or reset by the CCU in response to information received from the CPUs.
Frame Sync	- One of four patterns of pulses transmitted on the serial data bus. It consists of three positive pulses followed by three negative pulses and precedes the first word in every frame.
Hard Addresses	- Those addresses assigned uniquely by physical location to the Memory, DMA, and SIU Modules and to which those modules respond in the hard address mode.

Hard Address Mode	-	One of the two addressing modes in the BFTSC; used for reconfiguration and test purposes.
Hardware Status Words 1 and 2	-	Two words, accessible under program control, which contain information concerning the BFTSC's current configuration.
I/O Clamp	-	A signal from each Circumvention Module to all I/O modules which inhibits all I/O during a radiation event or power switchover.
Interrupt Mask	-	A set of signals in each CPU, set or reset under program control, each of which inhibits a different interrupt from being detected by the CPU.
Memory Clamp	-	A signal from each Circumvention Module to all Memory Modules which inhibits memory operation during a radiation event or power switchover.
Monitor	-	A device used to verify the operation of a specific function.
Monitor CPU	-	One of the two powered CPUs; verifies the outputs from the active CPU by comparing them to its own internally generated signals.
Monitor Mask	-	A set of signals in the monitor CPU, set under program control, each of which invalidates a different monitor generated signal for the purpose of verifying the operation of the monitor CPU.
NDRO	-	Non-destructive read out; contents of a memory location are not altered when that location is read.
Non-Volatile	-	A memory whose contents are not affected by a loss of power.
Normal Mode	-	The computational (non-reconfiguration) mode of the BFTSC.
Parity	-	See Error Codes

Peripheral Mask	-	A set of signals in each CPU, set or reset under program control, each of which inhibits a different I/O module's bus access requests and interrupts from being detected by the CPU.
Program Flag	-	Any of the seven three-bit patterns passed, under program control, from the active and monitor CPUs to the CCUs; used for testing and reconfiguration purposes.
Real Time Interrupt (RTI)	-	A signal generated by the active Timing Module every 8.8064 milliseconds in the BFTSC; causes an interrupt to the CPU and begins a serial bus frame in the SIU.
Reconfiguration Mode	-	Mode entered by the BFTSC following a fault; the normal mode is reentered only after the fault has been isolated and the faulty element replaced.
Refresh	-	The replacement of the contents of each location within one Memory Module with the corrected contents of that location in a different Memory Module (cross refresh) or with the corrected contents of that location in the same Memory Module (self refresh).
Rippler	-	Subelement switching device used in the Memory Modules (bit rippler) and at the address and data bus interfaces (byte rippler).
Simplex Mode	-	Mode entered only on external command; when the BFTSC is in this mode, all Category II faults are ignored.
Single Point Failure	-	A functional system failure caused by a single open or short circuit.
Soft Addresses	-	Those addresses assigned under program control to the modules in the BFTSC and to which those modules respond in the soft address mode.
Soft Address Mode	-	One of two addressing modes in the BFTSC; used for normal fetch and store operations.

Spare Module Redundancy	-	Redundancy at the module level; an entire module can be directly replaced by a spare.
Subelement Redundancy	-	Redundancy at the subelement level; an element can be divided into several identical subelements and failed subelements are replaced by spare subelements.
Syndrome	-	The EXCLUSIVE OR of the error code stored with the data word in memory and the error code computed for that word; points to the failed bit for single bit failures.
Test Zeros (Ones)	-	A function of each Memory Module; when in the test zeros (ones) mode, a Memory Module stores all zeros (ones) into any location written to and sets a status bit for every one (zero) in any location read.
Triple-Modular Redundancy (TMR)	-	Redundancy technique in which three identical, independent elements perform the same tasks simultaneously and whose outputs are subjected to a majority vote.
Watchdog Timer	-	Any of the several timers in the BFTSC used to monitor the elapsed time between two related events (e.g., the initiation and the completion of a bus transfer) and to generate an interrupt or a fault indication should this time exceed a specified limit.
Word Sync	-	One of four patterns of pulses transmitted on the serial data bus. It consists of three negative pulses followed by three positive pulses and precedes each word except the first word in every frame.
Write Protect	-	A function of each memory module; any 1024 word subblock can be directed, under program control, to ignore any attempt to store into it and to indicate any such attempt in the Memory Module's fault status register.